TECHNICAL REPORT
NATICK/TR-06/012

AD_____

TECHNOLOGY EVALUATION FOR PAINTABLE COMPUTING AND PAINTABLE DISPLAYS: RF NIXEL SEEDLING

by
Ara Knaian
David Greenspan
William Butera
Joseph Jacobson
and
Neil Gershenfeld

Massachusetts Institute of Technology Media Laboratory
Cambridge, MA 02139

April 2006

Final Report
June 2004 – September 2005

Approved for public release: distribution is unlimited

Prepared for
U.S. Army Research, Development and Engineering Command
Natick Soldier Center
Natick, Massachusetts 01760-5056

DISCLAIMERS

The findings contained in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

Citation of trade names in this report does not constitute an official endorsement or approval of the use of such items.

DESTRUCTION NOTICE

For Classified Documents:

Follow the procedures in DoD 5200.22-M, Industrial Security Manual, Section II-19 or DoD 5200.1-R, Information Security Program Regulation, Chapter IX.

For Unclassified/Limited Distribution Documents:

Destroy by any method that prevents disclosure of contents or reconstruction of the document.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
15-04-2006	Final	June 2004 - September 2005

4. TITLE AND SUBTITLE
TECHNOLOGY EVALUATION FOR PAINTABLE COMPUTING AND PAINTABLE DISPLAYS: RF NIXEL SEEDLING

5a. CONTRACT NUMBER
DAAD-16-00-R-0012

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Ara Knaian, David Greenspan, William Butera, Joseph Jacobson and Neil Gershenfeld

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Massachusetts Institute of Technology
MIT Media Laboratory
20 Ames Street, Room E15-414
Cambridge, MA 02139

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Defense Advanced Research Projects Agency, Microsystems Technology Office (R. Tulis, R. Reuss)
3701 North Fairfax Drive
Arlington, VA 22203-1714

U.S. Army Soldier systems Center, ATTN: AMSRD-NSC-TP-S (H. Girolamo)
1 Kansas St., Natick, MA 01760-5056

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)
NATICK/TR-06/012

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release: distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

A paintable computer uses thousands or millions of identical autonomous microsystems. The microsystems are fabricated using a high-volume batch process, mixed with paint and coated onto a surface. Self-assembling code allows the microsystems to act in concert to provide a useful function. MIT staff constructed and programmed a lab-scale prototype paintable display with 1000 semi-autonomous nodes. Also, MIT staff constructed a functional prototype demonstrating power distribution to randomly oriented millimeter scale semiconductor devices. MIT staff performed a series of basic engineering calculations to determine the feasibility of paintable systems with thousands of < mm3 microsystems, including how to power them, manufacture them, and provide communications between them. Based on these calculations, a paintable display, powered by a zinc-air battery for 8 hours, or externally powered by capacitive coupling, appears feasible in principle, with an estimated paint manufacturing cost of $0.40 / cm2 of 0.5 mm resolution color display paint.

15. SUBJECT TERMS

PAINT	DISPLAY	MICROSYSTEMS	FLEXIBLE DISPLAYS	COMPUTER ARCHITECTURE
NIXELS	COATINGS	COMPUTATIONS	ZINC-AIR BATTERY	COMPUTER PROGRAMS
CODING	NEARFIELD	SELF ASSEMBLING	SEMICONDUCTOR	PAINTABLE COMPUTING
GRAPHICS	ALGORITHMS	COMMUNICATIONS	CAPACITIVE POWER	VIRTUAL SUBSTRATES

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Henry Girolamo
U	U	U	SAR	81	19b. TELEPHONE NUMBER *(Include area code)* (508) 233-5483

TABLE OF CONTENTS

Page

TABLE OF CONTENTS (Continued)

LIST OF FIGURES

Figure **Page**

LIST OF TABLES

Table **Page**

PREFACE

The V1.0 Dense Nixel Array, V1.0 Raster Image Processor, and V2.0/V3.0 Technology Evaluation were developed by the Massachusetts Institute of Technology Media Laboratory (MIT Media Lab) under the program management of the Defense Advanced Research Projects Agency (DARPA), Microsystems Technology Office, Arlington, VA, (Robert Tulis, PM, Robert Reuss, PM).

The interim program was completed under the direction of the U.S. Army Soldier Systems Center, Natick, MA, U.S. Army Research, Development & Engineering Command, from June 2004 through September 2005, contract number DAAD16-00-R-0012. The purpose of the interim program was to demonstrate the feasibility of building displays and other devices using a new architecture, called paintable computing.

A paintable computer uses thousands or millions of identical autonomous microsystems. The microsystems are fabricated using a high-volume batch process, mixed with paint, and coated onto a surface. Self-assembling code allows the microsystems to act in concert to provide a useful function. Prior work on the RF Nixel Program is included in Natick Technical Report No. *NATICK/TR-04/018, August 2004.* [8]

The report represents the final report for the interim RF Nixel Program.

TECHNOLOGY EVALUATION FOR PAINTABLE COMPUTING AND PAINTABLE DISPLAYS: RF NIXEL SEEDLING

SUMMARY

Over the past 35 years, the semiconductor industry has increased number of transistors that can fit in a given area by a factor of 100,000, while simultaneously reducing the cost per transistor by the same factor. Using the same resources required to make one modern desktop computer, one could construct tens of thousands of sand-grain-sized computers. Each of these machines, which we call *paint particles,* would have internal power conversion, communications, sensors, and actuators. Thousands of paint particles could collaborate with one another to accomplish macro-scale tasks. Paintable systems would be adaptable, field-extensible, and highly robust to component failure.

The MIT Media Lab explored the technical feasibility of building paintable systems. We constructed and programmed a lab-scale prototype system. Using self-assembling code, we programmed the system to act as a display. Working together, the particles render and display a postscript-format image file. In addition, we constructed a functional prototype demonstrating power distribution to and operation of randomly oriented millimeter-scale semiconductor devices.

The MIT Media Lab performed a series of basic engineering calculations to determine the feasibility of paintable systems with 1 mm^3 paint particles. Particles can dissipate 10 mW heat, generate 6 J of electricity from internal zinc-air batteries or 1.5 J from internal combustible fuel. Photovoltaic cells provide 300 μW outdoors and 3.0 μW indoors. Paintable systems can store battery reactants in the paint binder; 6 J / mm^3 of binder can be stored, and diffusion is fast enough to transport the reactants to the particles. Reactive power transfer is an efficient method to transfer power to sparse, randomly placed particles. The available power from reactive transfer is proportional to V_{DD}^2: 100μW at 3.3V and 12 mW at 35V. Inter-particle communications is possible via optical, near-field, and far-field electromagnetic systems. Optical systems allow communication with very low area (sub-mm) particles, and 24 pJ/bit. Near-field electromagnetic gives precisely controlled neighborhoods, localization capability, and 37 pJ/bit. Far-field radio communication between widely spaced particles may be possible at 60 GHz; antennas that fit inside 1 mm^3 exist; complete transceivers do not. A 32-bit CPU uses less than 0.26 mm^2 die area, 256K x 8 SRAM uses 1.1 mm^2, and 256K x 8 FLASH uses 0.32 mm^2. III-V LED's may be fabricated on Si wafers using SiGe virtual substrates.

The MIT Media Lab selected technologies for a 17" diagonal, 640 x 480, paintable color display application. We propose to use a mixture of two kinds of particles: 1.0 mm rendering particles, with a microprocessor and memory, and 110 μm display particles, with simpler circuitry. Display particles use 50 μW for indoor-readable brightness and 336 μW for outdoor-readable brightness. Storage of zinc-air battery reactants in the paint binder allows 8 hour battery life for indoor use and 1 hour battery life for outdoor use. Reactive power distribution allows continuous operation from external power. The 300,000 paint particles required for this display could be manufactured for about $350.

1 INTRODUCTION

1.1 Paintable Computing

Over the past 35 years, the semiconductor industry has increased the number of transistors that can fit in a given area by a factor of 100,000, while simultaneously reducing the cost per transistor by the same factor. [1,2] A Pentium IV microprocessor, for example, has 18 million transistors, all within a few square centimeters.

This feat of manufacturing has been driven by the demand for higher and higher performance microprocessors. However, we believe that there is an even better application for this manufacturing technology than making large, fast microprocessors.

Using the same resources as are required to make one modern desktop computer, one could imagine making tens of thousands of less-capable sand-grain-sized computers. Each device would communicate with its neighbors; each device would have sensors and actuators; together they would collaborate to accomplish tasks. For example, devices with computing light emitters and computing could be mixed with paint and coated onto a surface to form a paintable display.

Devices built using this architecture would be extensible and reconfigurable in the field to an unprecedented degree. For example, a battery-powered paintable display could be spray-painted onto almost any surface. If, after some time, a larger or higher-resolution display was desired, more paint could simply be added.

Each paint particle would be a semi-autonomous unit, with on-chip energy conversion, communications, sensing, and actuation capabilities. These devices would be small, ranging from 100 μm to 1 mm in size, roughly the size of a grain of sand. The particles could be mixed with paint (or other materials) to imbue computation, sensing, and actuation to surfaces and materials.

1.2 Motivating Applications

Here is a short list of applications to motivate the study of how to build paint particles and how to engineer their interactions.

- **Paintable Displays.** Make particles with an LED as an on-board peripheral, mix them with a viscous medium, and paint them onto a surface. When the paint dries, the devices function as a display.

- **Programmable Matter.** Make particles with the ability to apply forces on their neighbors, through electric fields, magnetic fields, or chemical interactions. The result is a self-reconfigurable robot with millions of nodes. Possible uses for such a device include:

  1. A mobile robot that changes its shape from a spider to a snake to a wheeled vehicle as dictated by terrain conditions.

  2. A microscopic medical robot that travels through the bloodstream and the body to accomplish medical tasks, changing shape as needed.

  3. Structural elements that exert active feedback control to stabilize buildings and bridges.

  4. A wrench that deforms the shape of its head to fit any bolt encountered.

- **Programmable Electromagnetic Transducers.** Make particles with electric and magnetic field transducers. Working together, paint particles can cooperate to emit and receive electromagnetic waves, functioning as an antenna with programmable and self-organizing geometry. The paint and particles could be, for example, packaged into a paint ball and fired onto the deck of a ship, and then cooperate to sense some quantity and transmit radio signals to a satellite.

- **Holistic Data Storage.** Make paint particles without peripherals, and embed them into the materials used to make an airplane. The result is "black paint," a more robust substitute for the black box on an aircraft. After a crash, instead of having to find the black box, investigators could get successively higher fidelity information by recovering more and more mass from the aircraft, all of which would be infused with data-storing "black paint."

These examples are meant to illustrate that the paint architecture and its associated enabling technologies, if successful, would be transformative in many application domains, both military and commercial. We feel that microelectronic devices packaged in boxes are only the tip of the iceberg, regarding what can be done with mass-production mirofabrication technology.

1.3 Advantages of the Paint Architecture

- **Adaptability and Field-Extensibility.** A device made with paint particles has a self-organizing structure, so it can be made larger or smaller as needed, by adding more raw material. A paintable display could be made larger by adding more display paint to the surrounding area. A programmable matter wrench could be made larger by dipping it into a bucket of free programmable matter.

- **Robustness.** All of the elements in a paintable system are physically identical and dynamically self-organizing, so a very large number of component failures can be sustained before system failure.

- **Flexibility, Novel Form Factors.** Because paint particles are very small and their interconnections are wireless, they can be coated onto arbitrarily shaped surfaces, as well as surfaces that flex. To take one example, this would enable flexible and conformable displays.

1.4 Background

A great deal of research has been done in the computer science and information science communities to explore programming models and data fusion algorithms for paintable systems. This area of research is typically called Amorphous Computing [3] or Pervasive Computing.

The self-assembling code programming model that we are using was developed in an earlier phase of the RF Nixel Project, and is documented in William Butera's PhD Thesis, "Programming a Paintable Computer." [4]

Some related hardware work, also at the systems level, includes the pushpin computing platform [5], the Smart Dust Project [6], and the Alien Technologies nano-block display backplane [7].

A great deal of work is going forward with the goal of miniaturizing power sources, communications systems, sensors, and actuators. This work and its relevance to paint is the main subject of the V2.0/V3.0 Technology Evaluation section of this document.

1.5 RF Nixel Project Plan and Current Status

This RF Nixel project plan has three milestone prototypes, called V1.0, V2.0, and V3.0.

V1.0: Build a prototype paintable computer with 1000 nodes. This prototype may be constructed with standard electronics packaging technology, using packaged IC's, passive components, and printed circuit boards.

V2.0: Build a prototype paintable computer using the smallest commercially available components and leading-edge packaging and interconnect technology, using the energy conversion, communications, sensing, and actuation technologies selected for V3.0.

V3.0: Build a paintable system with paint particles that fit inside 1 mm^3, using full-custom wafer fabrication. Engineer a paint binder fluid. Dice the wafer into the binder, paint it onto a surface, and demonstrate a functioning system.

The current status of the RF Nixel project, relative to these milestones, is that the V1.0 prototype system is complete. We have also completed the V2.0/V3.0 Technology Evaluation, which has allowed us to select technologies and components for V2.0.

2 V1.0 DENSE NIXEL ARRAY

2.1 Materials and Methods

The V1.0 Dense Nixel Array, shown in Figure 1, contains about 1000 V1.0 paint particles, placed in a random array on a wooden board. Each particle contains an ARM7TDMI processor, 256KB of RAM, 2 MB of FLASH, infrared optical communications, an LED, a light sensor, and a heat sensor. For more information about the design of the V1.0 particles, see our previous final report. [8]

During the period covered by this report, we fabricated a second revision of the peripheral I/O card, assembled the 1000 particle array, built the power supply system, refactored the low-level software, implemented a distributed graphics engine using self-assembling code, and demonstrated functional postscript rendering on the system, shown in Figure 2.

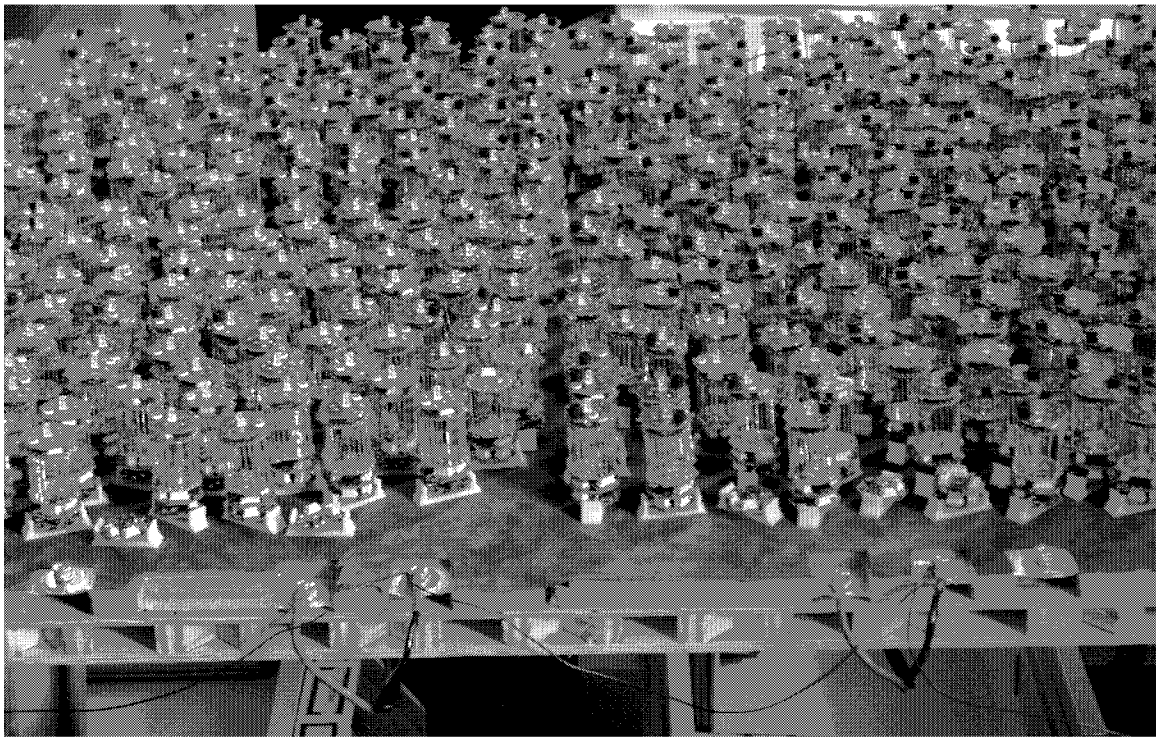

Figure 1. The V1.0 Dense Nixel Array

The V1.0 Dense Nixel Array contains about 1000 identical particles. Each particle has its own power conversion, communications, processor, and memory, as well as a multicolor LED, a light sensor, and a heat sensor. Power is distributed by two metal planes embedded in foam beneath the array, via a 'pushpin' mechanism. [8] The array is divided into eight power zones. Each zone is powered by 6VDC @ 20A.

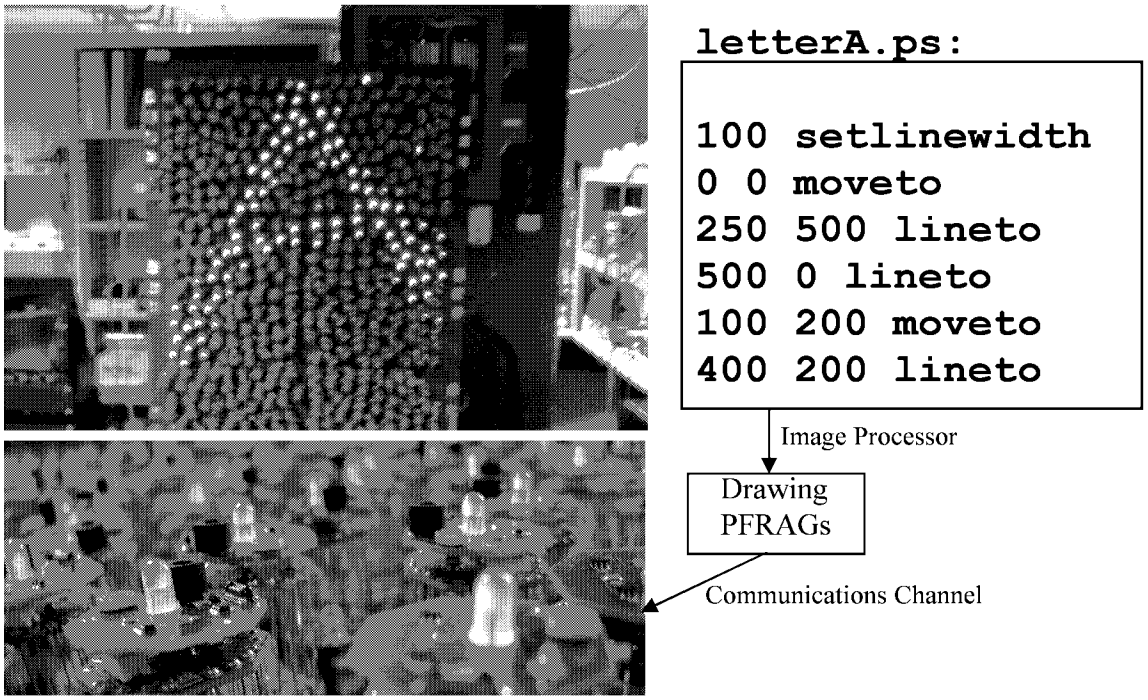

Figure 2. Distributed Postscript Rendering on the V1.0 Dense Nixel Array.
We built a fully distributed postscript rendering engine using the self-assembling code programming model. Postscript code is converted line-by-line into process fragments (small binary executables with associated data segments) which are injected into the system. The process fragments travel from node to node via optical communications. Each process fragment is responsible for drawing one of the objects specified in the postscript file. The nodes start out identical, without knowledge of their coordinates. After powering the system up, we inject a set of process fragments that allow each particle to determine its relative location, based on its connectivity graph and those of its neighbors. [8] Section 3 (page 10) of this report contains additional information about the rendering software.

2.2 Results and Conclusions

The performance of the V1.0 Dense Nixel Array was in line with our expectations. The algorithms that we tested on our simulator [8] worked well in practice. In the course of building the system, we did discover a few problems with the V1.0 design. In the interest of avoiding these problems in the V2.0 design, we summarize them here:

A. Defective Particles

About 20% of the particles did not work when we received them from our contract manufacturer. In the paint architecture, occasional bad nodes should be acceptable, since the system is self-organizing and assumes random placement and presence of functional particles.

However, in the case of the V1.0 system, one of the most common particle defects was for the infrared communications LED to be stuck "on." Particles with defective LED's flooded the communications channel in their neighborhood, preventing communication between other particles. In addition, a smaller fraction of particles had shorted power supply inputs. Particles with either of these defects had to be manually removed before the system would function properly.

Because it would be too expensive to 100% test particles at the V3.0 level, some defective particles are inevitable in a paintable system. However, we feel that paintable systems with defective particles can be made to work, if the following design principles are followed:

1. Particles should fail fast. Particles should perform a power-on self test. If any defect is detected, the particle should attempt to turn itself off.

2. Particles should be designed with defects in mind. When a particle is designed, a list of the particulars ways in which a particle might be defective should be drawn up. The design should be altered so that defects that might affect other particles become less likely.

3. The system should be designed with defects in mind. The communications system and power distribution system should be selected and designed to minimize the chance that a defective particle can interfere with the operation of other particles.

In the case of the V1.0 system, if reactive power distribution had been used instead of conductive power distribution, shorted power supply inputs would not draw down the system bus. If a carrier-modulated infrared system had been used, rather than an on-off keyed system, a stuck-on IR LED would not jam the communications channel. If the circuit traces leading to the IR LED were given more clearance to other traces, the stuck-on LED failure would happen less often. Finally, a fast-acting fuse at the power-supply input terminals would have prevented shorted particles from drawing down the power supply line.

B. Frequent Packet Collisions

The effective communications bandwidth turned out to be lower than anticipated, due to frequent collisions between packets sent by the particles. Several good solution pathways to make packet collisions less frequent exist; see Section 3.

C. Concurrency and Race Conditions

A paintable system is very unforgiving of bad programming practice, such as unchecked race conditions and memory leaks. Because there are 1000 processors running the same code, but with different inputs, any race condition that exists in

the code is bound to show up somewhere; in our experience, probably within the first minute after the system is powered.

From the perspective of building a demonstration system, this was a problem. But for an industrial software engineer, aiming to produce 100% reliable code, this would be a feature. Running code on a paintable system exercises it thoroughly in a short time. This makes it easy to reproduce intermittent problems, so that they can be identified and repaired before the software is deployed.

3 V1.0 RASTER IMAGE PROCESSOR

3.1.1 Introduction

PostScript is a computer language used to draw graphics. Called a page description language, it was originally created for use in laser printers. Although what a laser printer prints is essentially just a grid, or raster, of dots, a length of an inch contains hundreds of dots, and an entire page contains millions, so that it is impractical to simply tell the printer the desired color of each dot. One idea would be to give the printer a list of the coordinates of the lines, polygons, and other primitives to be drawn. However, drawing any kind of shape that was not anticipated by the maker of the printer might require sending hundreds of coordinates to the printer as an approximation. The answer is to send the printer the code that generates the coordinates.

PostScript makes this possible by being a programming language rather than just a data format. PostScript code is generated when a document is printed, and the code is executed by a processor in the printer called the raster image processor (RIP), which determines the raster of dots printed on the page.

The RIP for the paintable computer is structured a little differently. The input is a PostScript file on a desktop computer, but it is impractical for the desktop computer to prepare a raster to be sent to the paintable display. While in the case of laser printers this is precluded by the large number of dots, in the case of the paintable computer there are several other issues. The desktop does not know how many particles there are, how they are spaced or where they are positioned, and these quantities could be subject to change. Even if the desktop were somehow given this information, it has no way to address the particles individually.

The next idea would be to send a list of primitives, but again there is the problem of unanticipated shapes. It is compounded by the impracticality of preloading every particle with the graphics features of a laser printer. Once again the answer is to send code, not data, and in this case not to anticipate any specific shapes.

In this design, the desktop computer interprets the PostScript file and generates code in the form of process fragments (pfrags) to be injected into the paintable computer. The PostScript command to draw a line, for example, causes a Line pfrag to be injected. Since the paintable computer is not given the concept of a graphical line in advance, this pfrag includes a full mathematical description of what a line looks like. An injected pfrag could draw any shape or combination of shapes that can be computed by a computer.

3.1.2 Operation

Most of the work in turning the paintable computer into a display is establishing a coordinate system. When the paintable computer boots up, there is almost complete

symmetry between particles. A coordinate system needs to be specified, and the particles need to be able to determine their coordinates using a very limited amount of information; essentially what they have to work with is knowledge of which particles are neighbors, meaning that they are within communication range of each other.

All of the pfrags discussed here initially spread themselves so that every particle has a copy, but they may later delete themselves.

First a Display pfrag is injected. This pfrag uses the particles' LEDs to give visual feedback about the current state of each particle. During the set-up steps its color changes as new information is discovered. Once the particle has determined its coordinates, the Display pfrag defers to the graphics pfrags so that the image can be shown.

Three points define the coordinate system — one at the origin (O), one on the x-axis (X) and one on the y-axis (Y). Each of these points is assigned to one of the particles interactively. The way this is done is that for each of the three points, for example X, a pfrag is injected called GradientCarrierX, carrying inside it a pfrag called GradientX. (An encapsulated pfrag is just data until its carrier releases it, at which point it becomes a normal pfrag.) GradientCarrierX runs on every particle, watching the light sensor. When a flashlight is shone on one of the particles, GradientCarrierX on that particle unloads GradientX, and also puts a post on the homepage that triggers a mass extinction of GradientCarrierX. The result is that a single particle has been chosen as the X control point, and GradientX originates from this particle. This is repeated for O and Y.

A gradient is a paintable computing design pattern that allows particles to estimate their distance from a source. The particle that originates a Gradient pfrag (the source) is said to have a "hop-count" of zero. Each of the particle's neighbors sets its hop-count to one, and neighbors of neighbors acquire a hop-count of two. The rule is that each particle sets its hop-count to one more than the smallest of its neighbors' hop-counts. Each particle then averages its neighbor's hop-counts to compute a value that is approximately proportional to its distance from the source.

The Distance pfrag, injected next, runs on each particle waiting for gradient values corresponding to the three control points. It calculates the distance to each point, and posts the information to the homepage. It also performs the task of measuring and propagating the values of the distances OX, OY and XY.

The Coordinates pfrag watches for posts from the Distance pfrag. Calling P the location of a given particle, once the Coordinates pfrag sees values for PO, PX, PY, OX, OY, and XY, it uses geometry to calculate its own coordinates.

Once each particle knows its coordinates, the paintable computer becomes a canvas ready to be drawn on. Graphics pfrags can be injected which will perform calculations based on each particle's coordinates to determine how to affect the color of the particle's LED. In the paintable display demonstration, client software processes a PostScript file and injects a stream of Line pfrags that form a shape.

3.2 System Architecture

3.2.1 Operating System

Particles run a custom operating system that is downloaded into their Flash memory before they are used. This OS was developed using the ARM developer tools. Start-up code is written in ARM assembly language; code that manages the hardware peripherals is written in C; and higher-level code having to do with homepages and pfrags is written in C++. Using C++ over C facilitated development without substantially affecting code size and performance.

Each particle carries a unique particle ID; this is the only asymmetry between particles. The ID is used in data transmissions so that the series of messages sent by a single particle over time can be identified as such by receiving particles. Each particle's ID was chosen at the time of the particle's first programming.

All communication (particle-particle and particle-desktop) uses the IP protocol. For communication with the desktop over a serial connection, SLIP is used to transport the packets. Interparticle communication is continual and asynchronous, and thus requires packet queues that dynamically allocate memory, as well as double buffering.

The desktop computer may control particles over the serial connection individually or in parallel using what is called the "monitor" interface. The monitor is used for maintenance, testing, and pfrag injection. An OS upgrade can be performed on all particles at once using suitable monitor commands; the OS is sent from the desktop in chunks which are received by each particle, assembled, check summed, and written to Flash memory. Particles can also be told to change the color of their LED or delete a pfrag, for example.

The mechanism used by the monitor is a simple stack-based language. Each bytecode in the instruction set may push onto the stack a four-byte integer or an arbitrary-length chunk of data (the two data types) or may operate on the existing stack. This provides an abstraction over all monitor functions, which take varying numbers of arguments, some of which are large blocks of data, and also allows arithmetic expressions of a particle's ID or other properties to be evaluated and used as arguments or as conditions on execution.

The monitor client software that runs on the desktop computer during operation is written in Java, using a cross-platform serial I/O library.

3.2.2 Homepages

The implementation of homepages follows Butera's definition closely. [8] A homepage is a collection of posts. Each post consists of a pfragID (the pfrag responsible for the post), a postID (the pfrag's "name" for this piece of information), and arbitrary content data.

A particle's entire homepage is periodically broadcast for the benefit of all its neighbors. Each particle maintains local copies of its neighbor's homepages as they were last received. Pfrags acquire information by perusing neighbor homepages; in actuality they access local mirrors that are automatically updated. Pfrags can send information only by posting to their host particle's own homepage, or removing a post from it; the homepage is then broadcast automatically.

There are some design decisions regarding the presentation of the homepage API to the pfrag. For example, as a pfrag iterates over the neighbor homepages in residence, should it have access to identifiers that allow it to keep track of specific homepages over time, or should it be presented with a set of unlabeled homepages that may or may not belong to different particles than during the last iteration? As another example, should a pfrag be able to modify or remove posts belonging to a different pfrag? In our test applications homepages were presented to pfrags as unlabeled, and pfrags did not delete other pfrags' posts.

3.2.3 Pfrags

A process fragment is a unit of mobile code that can migrate between particles. Programming a paintable computer at the application level means constructing a set of pfrags that interact to produce the desired behavior. A pfrag accomplishes its work by calling into the OS using the pfrag API, which allows it to perform several kinds of actions:

- Setting the LED and reading sensors
- Adding, removing, and reading posts on its own homepage
- Iterating over neighbors and reading posts on neighbor homepages
- Copying itself to neighbors (migration)
- Deleting itself
- Accessing a pfrag parameter
- Calling math functions, retrieving random numbers
- Releasing an encapsulated "subfrag"

Pfrags are implemented by passing machine code as data. Each pfrag is compiled on the desktop computer to a binary image, and this image becomes the pfrag (though post-processing may be performed as described below). Included in the header is information about how the code is divided into sections, such as read-only code and data, read-write data with initial values, and read-write data that is initially zero. These sections must be

treated differently. For example, the zero-initialized data section does not appear in the pfrag at compile-time but is allocated at runtime, while the initialized data section is present at compile-time but can change at runtime such that the initial values are lost.

We decided that pfrags should maintain their state when they migrate. This allows a pfrag that is copying itself to a neighbor to control the new copy's state when it arrives, by temporarily altering its state for the duration of the copy operation. It would be a small change to create a system where copied pfrags are reinitialized, or even one where pfrags can selectively "reboot" their state, simply by always retaining a clean (unmutated) copy of the initial values of the read-write data.

Encapsulated "subfrags" were invented while developing the paintable display application. For this application, gradients must be originated in response to sensory input. It made sense in the interest of modularity to separate the gradient from the mechanism for instigating the gradient. And in designing a gradient pfrag for a system where pfrags maintain state across migrations, it is most intuitive and general to assume that the code is in its initial state at the source particle of the gradient. The solution that was hit upon was to allow pfrags to "inject" other pfrags into the current particle. This was implemented by adding an optional "subfrag" section to the pfrag definition. This section acts as read-only data inside a pfrag acting as a "carrier" until the carrier decides to release it. It is then stripped from the carrier pfrag and comes into being as an independent pfrag. On the desktop client, individual pfrags can be compiled and then nested as desired as a post-processing step before injecting the outermost pfrag into the paintable computer.

Another new concept that came out of this work is that of a parameterized pfrag. When a parameterized pfrag is designed, its number of parameters is specified, and each one gets a spot in the header structure of the pfrag, rather than being mapped arbitrarily by the compiler. This means that values can be assigned to a pfrag's parameters as a post-processing step after compilation without delving into the pfrag machine code or recompiling. In the paintable display, the Line pfrag is parameterized to allow the desktop client to create pfrags representing lines with arbitrary coordinates without performing any recompilation.

3.3 Conclusions

3.3.1 Efficiency

This system as implemented is inefficient in a number of ways concerning data representation and data transmission, described below.

3.3.2 Communications

No mechanism is used to detect or prevent packet collisions. Particles merely send their broadcasts with pseudorandom timing to avoid being locked into unfavorable

transmit schedules. Ultimately the overall data rate is limited by the number of packet collisions, where a collision means any overlap in time between two packet transmissions from particles within communications range, in which case both packets are discarded by all recipients within range of both transmitters. To bring data rates closer to optimal (what could be achieved if an external "traffic director" could tell the particles when to transmit), better feedback and negotiation is needed.

Methods for negotiating use of a channel are an active research area. It is of great importance what assumptions are made about the communications hardware. For example, one set of assumptions leads to Ethernet, a time-tested standard that allows devices to share a medium effectively by having each device watch for collisions and adjust its behavior without further negotiation.

Another source of inefficiency at the communications layer is the system of buffers and queues used to handle incoming packets, mostly with regard to the memory consumed in RAM. However, it is not clear that the system of handling packets can be much reduced in complexity. A surer route to reducing the memory usage is simplifying the homepages and pfrags that must be stored and transmitted.

3.3.3 Homepages

Homepages are a simple way to manage the sharing of information between particles. They successfully abstract away the sending and receiving of information so that it is not under direct control of the pfrags, but they also result in more information being transmitted than is theoretically necessary.

No distinction is made between privately useful and publicly useful information; all information is broadcast to neighbors. Many posts in the paintable display system were only meant for within-particle communication, but they consume both bandwidth and memory by being broadcast to and mirrored by neighbors. This could be remedied by introducing a distinction between public and private.

Information is broadcast whether or not it has changed. However, this is a complicated issue, because packets can be dropped and particles can lose and regain contact. The idea of transmitting only changed information wrongly assumes that the state of the receiver is known to the sender

.

Information is broadcast whether or not any receiver is interested. Addressing this problem would address both of the above. Consider one possible model. Particle A expresses interest in data x, and continues to express interest periodically until Particle B provides x. When Particle B receives A's message, Particle B periodically broadcasts x, remembering that it is fulfilling A's request, until A no longer requests x. This method is robust with respect to dropped packets, and if there are no dropped packets then neither particle waits. Much would remain to work out, however; for example, how information is labeled, how requests and responses work with multiple neighbors, what network layer

this mechanism should operate at, and how infrequently-changing information can be spread efficiently.

3.3.4 Pfrags

Pfrags take up a lot of particle RAM and some communication bandwidth. In the current architecture, pfrags such as Gradient and Line can occupy as much as a kilobyte each with their code and state. Given the conceptually simple functions they perform, there is the question of in what way they could be made smaller. There are also the questions of how to be more efficient with their storage and transmission.

One idea is to separate the code and read-only data of a pfrag from its state. As motivation, consider a scenario where three different Gradient pfrags are running on a particle, each originating from a different source. The particle loses contact with its neighbors, and the three pfrags delete themselves because they assume that the gradients are no longer active (whether or not this really happens depends on different aspects of the design and the situation). Then the particle establishes contact again, and its neighbors notice that it is lacking the gradients. The neighbors go into "spread" mode and retransmit the pfrags, so that all three are retransmitted, possibly from multiple neighbors.

Now imagine a change in the architecture such that the first time the particle receives a gradient, the code is placed in a cache. This code is then shared by all three gradients. If the Gradient pfrags delete themselves, the cached code remains as long as memory allows; it contains no state and does not indicate the presence of a pfrag, it is merely for reference. Then Gradient pfrags re-spread by neighbors in the future do not need to include the gradient code, just state information. This could save a lot of RAM and bandwidth. The challenge is that particles containing a pfrag that wishes to transmit itself do not necessarily know whether the recipients need copies of the code. Perhaps particles would inform their neighbors of all the pfrags for which they have code, or would be required to request copies of the code they don't have.

This optimization would allow a message of just a few bytes sent between particles to be a pfrag. If that kind of pfrag is very common, it is likely that the receiver would already possess the code.

Ultimately, however, pfrag code needs to be made much smaller. From an information theoretical standpoint, there is not much information in a 500-byte Gradient pfrag. There is identifying information such as a pfrag ID. There is code to manage spreading to uninfected neighbors, but most pfrags have almost identical code for this. There is the information that the hop-count homepage post should be set to one less than the minimum of the neighbor's hop-counts. Finally, there are mechanisms for allowing the gradient to be dissolved without re-spreading, which will be discussed in the next section.

A promising route to drastically cutting down on the size of pfrags is to use a more mathematical programming language. The hop-count of a pfrag is like a variable. A vector containing all of a pfrag's neighbors' hop-counts would be made easy to specify in the pfrag programming language, and then a "min" operation can be performed on this vector. This is much less code then the "for-loop" in C currently used to perform this operation. The other part of this approach is building more intelligence into the architecture. In the current architecture, the pfrags are very procedural, and the operating system exposes a small set of tools without really knowing what the pfrag is trying to accomplish. If pfrag behaviors such as spreading to all particles or keeping a piece of information up-to-date were understood by the operating system, then a pfrag could take on a more compact, declarative form. This simultaneously reduces the burden on the pfrag designer, or the designer of a system that generates pfrags automatically.

3.3.5 Robustness

Overall, the homepage/pfrag architecture succeeded in creating a paintable computer on which applications can be written that function correctly despite the asynchrony and unreliability of communication between particles. The basic homepage model in particular, despite the inefficiencies described above, was easy to use and avoided timing issues in data communication. However, there were some unforeseen challenges in accomplishing correct behavior with pfrag migration, especially when trying to create a pfrag that would spread to all available particles and later undergo mass extinction.

Gradient pfrags, in particular, were designed to propagate across the field of particles but be ready to delete themselves when they no longer detect a pfrag with smaller hop-count. Since the gradient source is the only particle with a hop-count of 0, its deletion is meant to eventually trigger the deletion of all occurrences of that particular gradient. In reality, though, with dropped packets and particles that become congested and temporarily unresponsive, there can be dynamics where gradients re-spread just as fast as they are deleted. During attempted extinction, as long as a Gradient pfrag momentarily has support from a neighbor with a smaller hop-count, it can spread back to neighbor particles that have already deleted the gradient. Even the piece of information that some neighbor has a smaller hop-count could be out of date by the length of a homepage update period, or longer if a homepage packet was dropped.

One technique used in the paintable display to counter this problem is for the gradient source to generate a sequence number that increases over time and propagates outwards; in other words, the source periodically increments its sequence number, and each Gradient pfrag that is not at the source watches for a neighbor with a higher sequence number and brings its own up to date. Each time the sequence number is increased, the pfrag has a chance to copy itself to neighbors, and this is the only time it is allowed to. When the gradient source pfrag is deleted, there is a limit on spreading while the other Gradient pfrags are being deleted, because no globally new numbers are generated. If a pfrag's sequence number is n less than the global maximum (the last number generated by the source before it was deleted), it can spread at most n times in the worst case.

More generally, trying to delete a pfrag from the entire paintable computer when that pfrag is programmed to spread can be difficult. To allow a pfrag to be deleted from the entire computer, either the pfrag must be programmed so that it is dependent on its neighbors for its survival, or there must be an active agent such as another pfrag that ensures the deletion of the first pfrag. However, then there is the problem of deleting the second pfrag. This dependency on neighbors may be natural for some applications, however, as it is for the gradient. For example, in mathematical problem solving a pfrag might delegate a subproblem to a neighbor particle. The neighbor pfrag then reports the answer back to the original pfrag, or deletes itself if there is no one to report back to.

In refining the paintable computer architecture as applied to wireless devices, it is suggested that packet loss and communication delays be considered the norm rather than the exception, and that the assumptions made in this regard be carefully examined. For example, in the original proposal for paintable computing, a communications model with zero packet loss, but unbounded delay, was assumed, the rationale being that unreceived packets can always be resent automatically, and it is unlikely that a large number of resends will be required. However, when all transmissions are heard by all neighbors and state is constantly changing, it is hard to have an effective resend policy. Zero packet loss may be a simplifying assumption for the higher levels of the architecture, but including the lower levels it may be a complication, since packets in reality are lost all the time, and transmit time is precious.

It should be remembered, however, that this all depends on the hardware implementation. In a wired system, packets are not generally lost. In a wireless system, it depends on the capabilities of the communication hardware, and also in what ways the connectivity of the network may change while the particles are running.

3.3.6 Development & Debugging

Debugging the hardware and software of the paintable computer was a challenging aspect of the project. Hardware could fail in many ways, from solder bridges to bad seating on the pushpin board, and it was time-consuming to locate defective particles given that they could affect their neighbors in various ways. However, the software was ultimately more challenging to debug, because most of what was going on in the system was hidden.

High-level simulation of the system had been performed beforehand to guide the design of the paintable computing platform, but there was no lower-level simulation to guide the design of the operating system. We are currently building a low-level simulator for a paintable computer. This simulator will simulate processor use at the instruction level and communication at the byte level, including overhead imposed by the operating system.

4 V2.0/V3.0 TECHNOLOGY EVALUATION

This section contains the results of the V2.0/V3.0[1] Technology Evaluation.

Before beginning the design of the V2.0 particle, DARPA and the MIT Media Lab felt it prudent to do some basic engineering calculations relevant to the design of paintable systems. These calculations are meant to answer questions like the following:

- How much power is available from various conceivable sources?
- How much power is required?
- How much heat can be dissipated by a particle?
- What kind of inter-particle communications system should be used?
- How much energy per bit is required for inter-particle communications?
- How will particles be manufactured?
- How much will it cost to manufacture particles?

This section contains the results of this study. We feel that it is a useful resource for anyone designing a system using the paint architecture. It contains charts, equations, typical values for parameters, literature references, and analysis techniques useful in testing paint applications for 1st order physical feasibility, and in the early stages of the design of a paintable system.

At the end of this section, we use the results of the study to make a case for the 1st order engineering feasibility of a 17" 640 x 480 paintable color display, and to estimate its manufacturing costs.

[1] See §1.5 for definitions of the terms V1.0, V2.0, and V3.0.

4.1 Heat

Heat dissipation is a critical factor in the design of particles containing actuators, light emitters, or high-performance computers.

Heat dissipation per volume increases as length scale goes down. This is because heat dissipation is approximately an area effect, and surface area to volume ratio goes up as length scale goes down. From the paintable system designer's perspective, this is the good news. It means that breaking a system into many small pieces and spreading them around allows it to run cooler.

However, the amount of heat that a single particle can dissipate certainly does go down as the length scale of the particle goes down. This section shows how much heat is cause for concern at a given length scale, from 10μm through 10 cm. Engineering charts of power vs. length scale vs. particle temperature are presented, for continuous and pulse-mode operation. Using these charts, the designer can get an idea of how much power dissipation is reasonable at a given length scale.

These charts are drawn based on a model for natural convection around a sphere. In other words, they do not take the geometry of the particle into account. Heat fins, actuators to blow air around, and other engineered heat transfer solutions can increase the allowable heat dissipation beyond that predicted by the charts in this section. Still, these charts can be used for setting specifications for a first-pass design.

4.1.1 Heat Transfer and Thermal Resistance

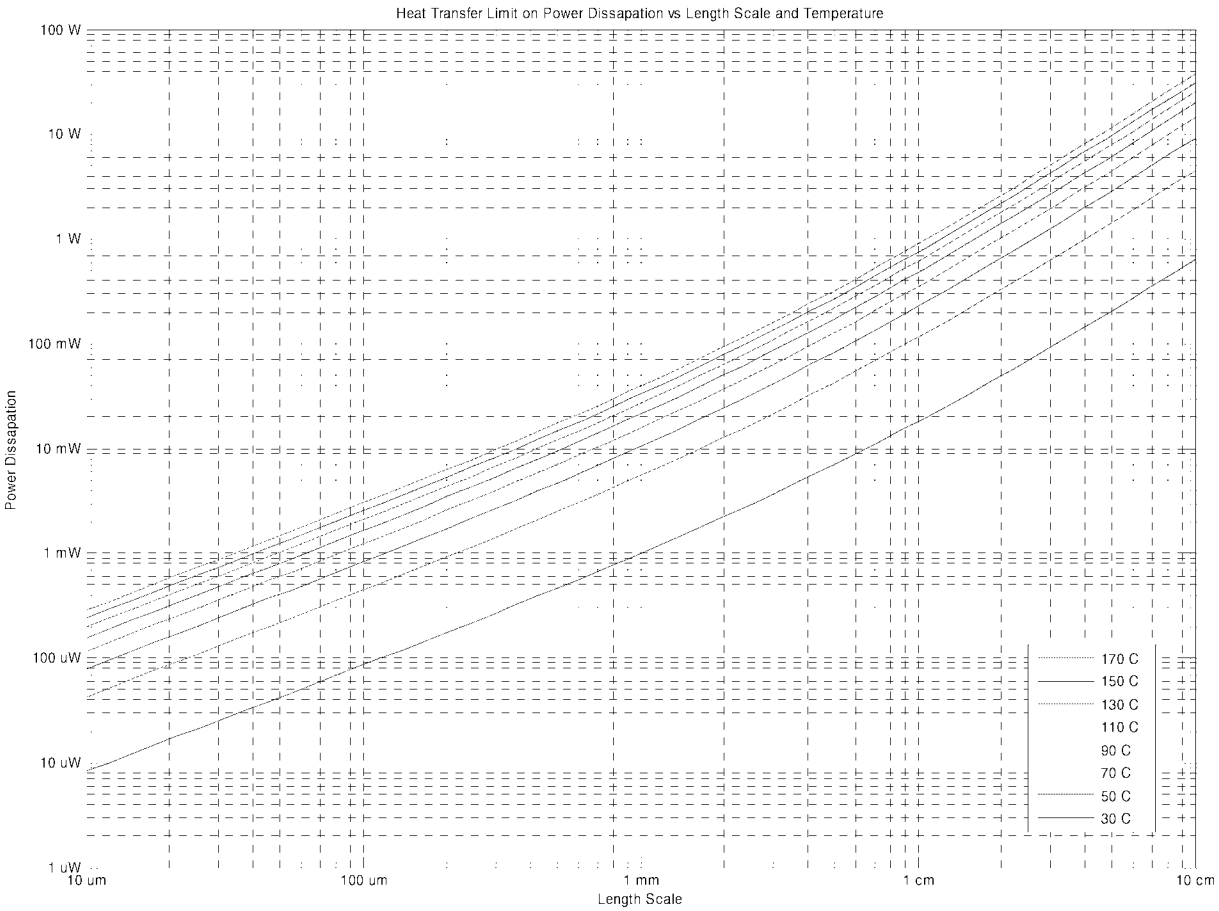

Figure 3. Heat Transfer Limit on Power Dissipation

This chart can be used to determine how much power dissipation is reasonable, given a length scale and maximum operating temperature. These are approximate values; see text for detail. From the plot, one can see that a 1-mm device at 70°C can dissipate about 10mW. (A 25°C free-air temperature is assumed for this plot.)

Figure 3 gives a first approximation for the steady-state operating temperature of a particle given its power dissipation and size. For example, using the table, one can see that a 1-mm device can dissipate up to 10 mW before reaching the 70°C standard maximum operating temperature for electronics, whereas a 100 µm device can only dissipate about 800 µW. Figure 2 shows the same data presented in thermal resistance format.

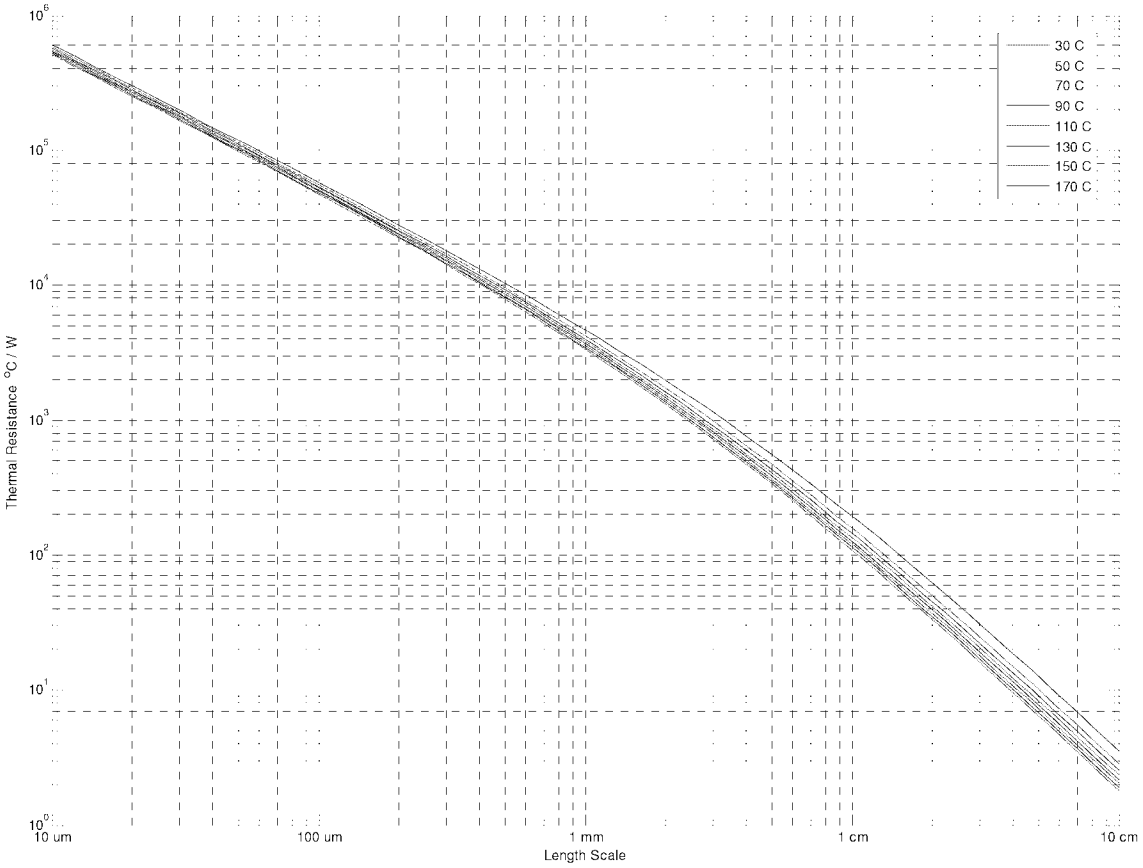

Figure 4. Thermal Resistance

This is the same information as Figure 3, presented in thermal resistance format, which can be more useful for calculation. These are approximate values; see text for detail. From the plot, one can see that a typical 1-mm device has a thermal resistance of about 4000 °C/W.

Figure 3 and Figure 4 show the heat transfer by natural convection and radiation from a sphere of a given diameter, at a given temperature. It is possible to achieve more heat transfer than shown by using a higher surface area shape than a sphere, increasing the surface area with fins or by adding actuators to move air. It is also possible to get lower heat transfer, by using a low surface area shape, or by coating the particle with a material that is not sufficiently conductive of heat.

Table 1. Calculating Heat Transfer and Thermal Resistance

$T \quad (K)$	Sphere surface temperature
$D \quad (m)$	Sphere diameter
$T_{amb} = 298 \quad K$	Free-air temperature
$T_{film} = \dfrac{T + T_{amb}}{2}$	Film temperature
$\beta = \dfrac{1}{T_{film}}$	Coefficient of volumetric expansion (for an ideal gas)
$g = 9.81 \quad m/\sec^2$	Acceleration of Gravity
$\nu = fcn(T_{film}) \approx 2 \times 10^{-5} \, m^2/\sec$	Kinematic Viscosity of Air
$Gr_D = \dfrac{(bouyancy)(inertia)}{(vis\cos ity)^2} = \dfrac{\beta(T - T_{amb})gD^3}{\nu^2}$	Grashof Number
$\Pr = 0.71$	Prandtl Number of Air
$Ra_D = \dfrac{(bouyancy)}{(vis\cos ity)} = Gr_D \cdot \Pr$	Raleigh Number
$Nu_D = 2 + \dfrac{0.589 \cdot Ra_D^{1/4}}{\left[1 + (0.469/\Pr)^{9/16}\right]^{4/9}}; Ra \le 10^{11}; \Pr > 0.5$	Nusselt Number Correlation, "Natural Convection on a Sphere"
$k = fcn(T_{film}) \approx 0.26 \quad W/m \cdot K$	Thermal Conductivity of Air
$h_c = \dfrac{Nu_D k}{D}$	Convection Heat Transfer Coefficient
$A = \pi \cdot D^2$	Surface area of a sphere
$\dot{Q}_c = h_c A(T - T_{amb})$	Convection Heat Transfer Rate
$\varepsilon = 0.80$	Emissivity of Sphere (Value for black anodized aluminum; typical value for non-reflective engineering materials)
$\sigma = 5.670 \times 10^{-8} \quad JK^{-4} m^{-2} s^{-1}$	Stefan-Boltzmann Constant
$\dot{Q}_r = \varepsilon A \sigma (T^4 - T_{amb}^4)$	Radiation Heat Transfer Rate
$\dot{Q} = \dot{Q}_c + \dot{Q}_r$	Total Heat Transfer Rate
$R_{th} = (T - T_{amb}) / \dot{Q}$	Total Thermal Resistance

Formulas and constants used in Table **1** are taken from [9].

4.1.2 Thermal Capacitance and Thermal Time Constant

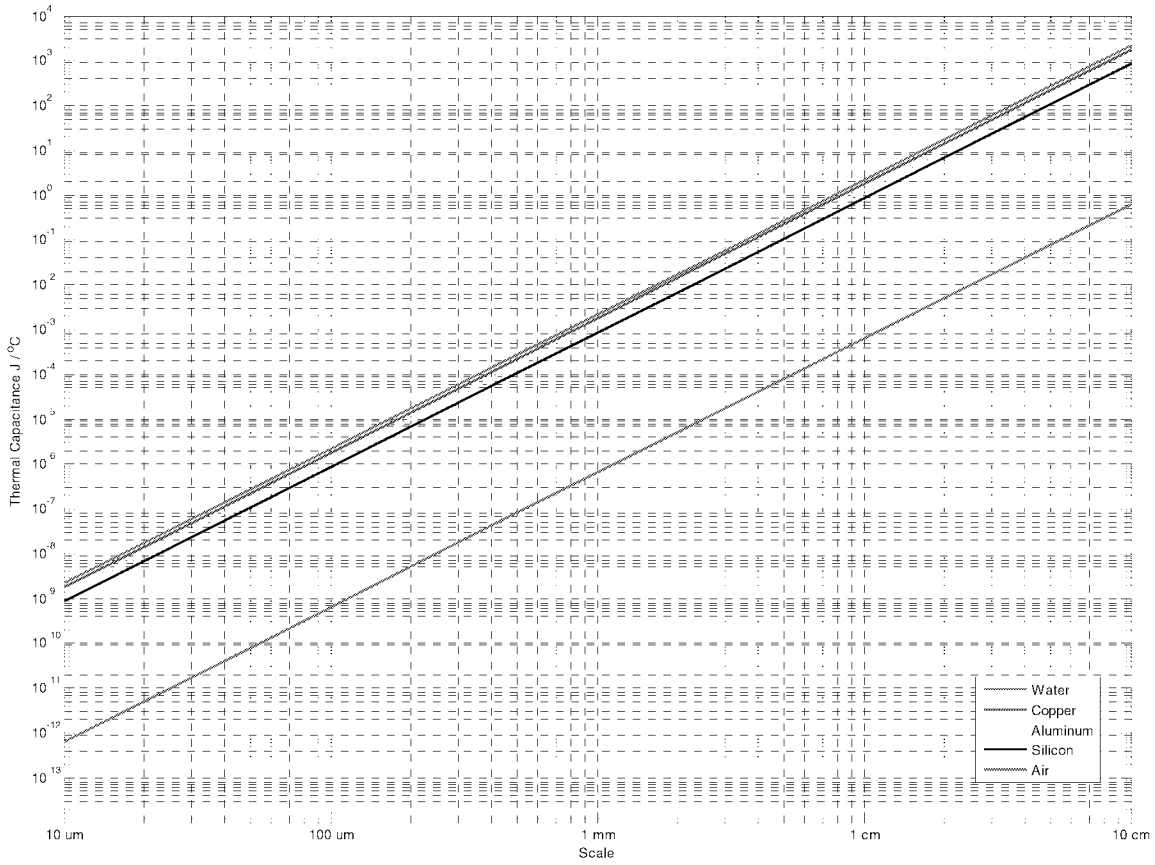

Figure 5. Thermal Capacitance

This chart may be used to estimate the thermal capacitance of a paint particle. It shows the thermal capacitance of a sphere with a given diameter, for a variety of materials. From the plot, we see that a 1-mm silicon device has a thermal capacitance of about 2 mJ / °C.

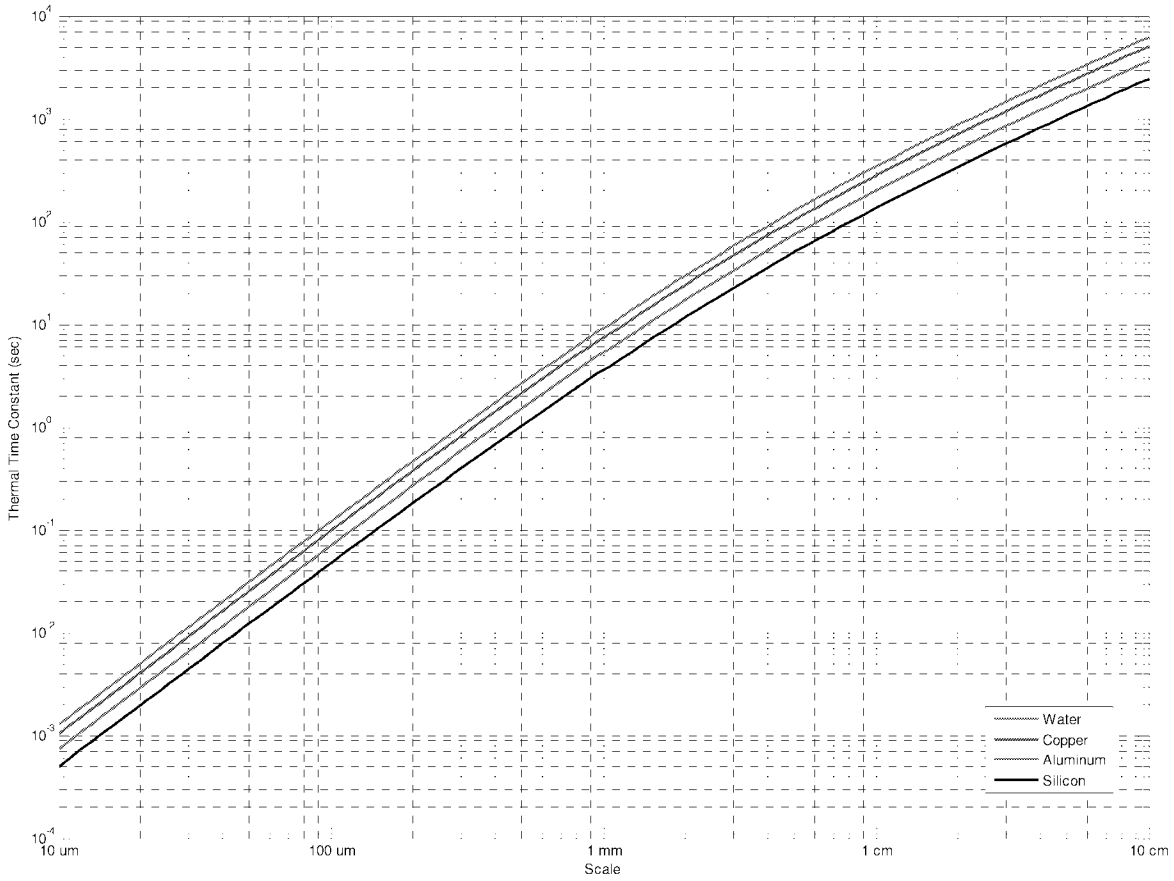

Figure 6. Thermal Time Constant

This chart may be used to estimate the thermal time constant of a paint particle. This chart is subject to the same approximations as Figures 1-3: that the system is a sphere of a given diameter. From the chart, we can see that a 1-mm system has a thermal time constant from 3-8 seconds.

Figure 5 and Figure 6 provide a way to estimate the thermal capacitance and thermal time constant of a paint particle. The thermal capacitance is the amount of thermal energy that a system can store per degree increase in temperature. The thermal time constant is a measure of the rate at which a system cools to ambient temperature. These are important figures in the design of a system that dissipates power in a pulse mode.

Table 2. Calculating Thermal Capacitance and Time Constant

R_{th}	Total Thermal Resistance
$V = \frac{4}{3}\pi \cdot D^3$	Volume of a sphere
$\rho_{Cu} = 8920 \quad kg/m^3$ $\rho_{Si} = 2330 \quad kg/m^3$ $\rho_{H_2O} = 1000 \quad kg/m^3$ $\rho_{Al} = 2700 \quad kg/m^3$	Density of Materials
$H_{Cu} = 380 \quad J/kg-K$ $H_{Si} = 705 \quad J/kg-K$ $H_{H_2O} = 4186 \quad J/kg-K$ $H_{Al} = 900 \quad J/kg-K$	Specific Heat Capacity of Materials
$C_{th} = \rho VH$	Thermal Capacitance
$\tau_{th} = R_{th}C_{th}$	Thermal Time Constant

4.1.3 Pulsed Power Safe Operating Area Curves

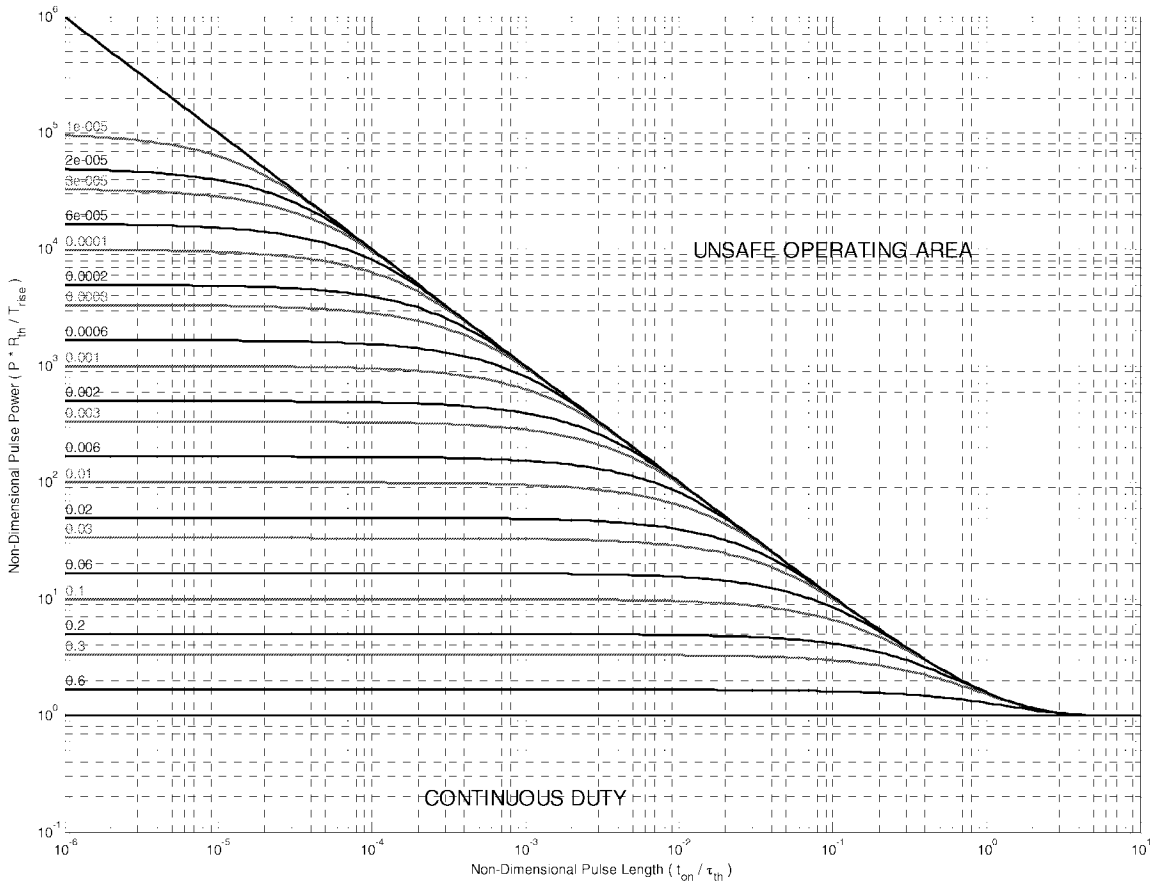

Figure 7. Thermal SOA for Pulsed Operation (Non-Dimensional)

This chart shows the generalized relationship between pulse length, thermal time constant, pulse power, thermal resistance, maximum temperature rise, and maximum duty cycle. In the lower region, continuous operation is allowed; in the upper region, a single pulse will cause the system to overheat; in the middle region, operation is allowed so long as the duty cycle is equal to or lower than that indicated.

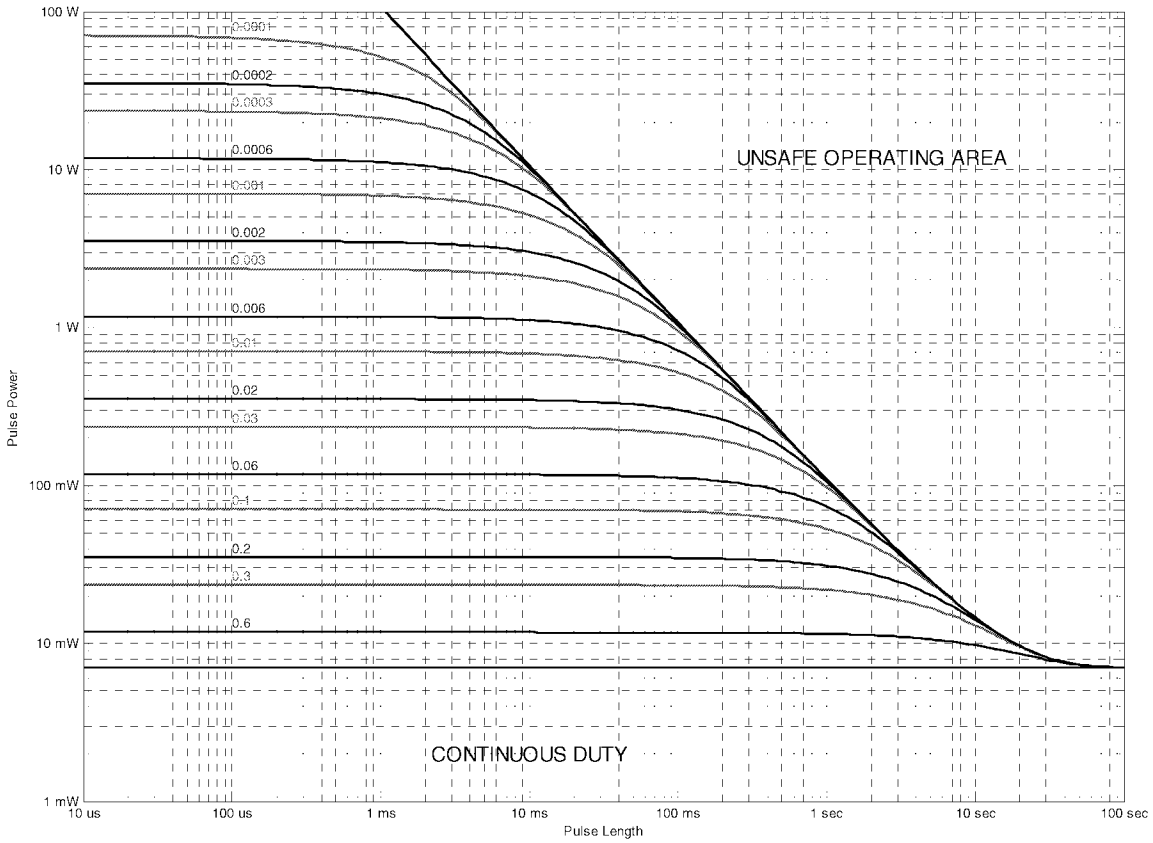

Figure 8. Thermal SOA Plot for Pulsed Operation, 60°C rise, 1-mm

This chart shows the maximum duty cycle for pulsed power dissipation from a 1-mm paint particle. Reading the chart, one can see that a 1-mm particle operating with 1-sec 30mW pulses (e.g. from an RF transmitter) must cool down for about four seconds between pulses (i.e. a duty cycle of 20%) to avoid overheating. This chart is drawn for a copper sphere in free air experiencing uniform volumetric heating.

Table 3. Calculating Thermal Safe Operating Areas

T	Particle Temperature
T_a	Ambient Temperature
T	Time
P(t)	Instantaneous System Power Dissipation
$\frac{dT}{dt} = C_{th}\left[P(t) - \left(T - T_a\right)R_{th}\right]$	Differential Equation describing particle thermal model
P_{pulse}	Pulse Power
t_{on}	Pulse Length
DC	Duty Cycle
R_{th}	Particle-to-Ambient thermal resistance
τ_{th}	Particle thermal time constant
$\frac{P_{pulse}R_{th}}{\Delta T_{rise}} = \frac{1 - e^{-(t_{on}/DC)/\tau_{th}}}{1 - e^{-t_{on}/\tau_{th}}}$	Solution to differential equation describing system model: relationship shown in Figures 5 and 6

4.2 Power

Physically, a paintable computer consists of autonomous microsystems, called paint particles, which are suspended in a fluid or gel, called the paint binder. The particles require power to function. The ways that one might provide power to the particles can be divided into three categories:

1. Particle-Stored Energy: The energy to operate a particle over its lifetime is stored inside the particle. The energy source is never replaced: once depleted, the system is discarded.

2. Binder-Stored Energy: The energy to operate the particles is stored in the paint binder. Each particle contains an energy conversion device, but the binder contains the energy source.

3. External Power: The power required to operate the particles is continuously supplied to the system, from an external, usually macroscopic source.

A very large fraction of systems made or sold today use external power, since they can almost always be plugged in, refueled, or have their batteries replaced. For example, automobiles, laptop computers, and homes all fall into this category. Examples of systems that use something akin to particle-stored power include rockets to lift payloads into outer space, glow-sticks, and cheap disposable flashlights with non-replaceable batteries.

If operation is a random environment is desired, the most straightforward solution is to store energy inside each particle. However, the economics of wafer fabrication dictate that cost is proportional to area, regardless of the complexity of that area. Because power sources tend to be large compared to other system components, the particle-stored energy solution places serve restrictions on at least one of the following: device power, device lifetime, and device cost.

Binder-stored energy decouples the volume of the particles from the volume of the energy source, by storing the energy outside of the particles, but still inside of the paintable system. This solution allows relatively high-power, long-lifetime, low-cost systems, compared to the particle-stored power solution. Some materials development will be required to implement binder-stored energy; we know of no prototypes or demonstrations of this concept to date. However, it appears possible in principal.

When continuous operation is required, external power is the only feasible solution. There are some external power solutions that may work in a random environment, (e.g. photovoltaic cells) however, most require a structured environment.

4.2.1 Particle-Stored Energy

4.2.1.1 Electrochemical Cells

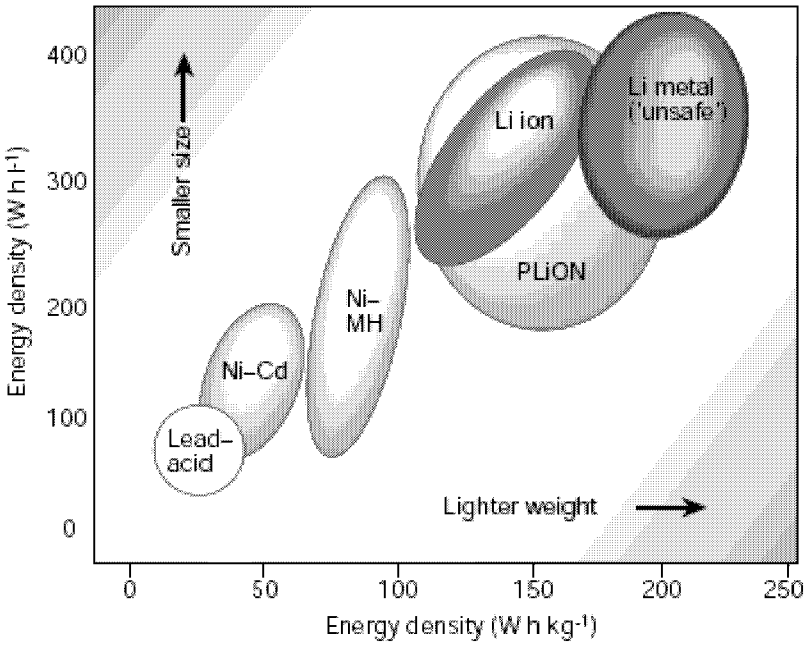

Figure 9. Energy Density of some Battery Chemistries (from [10])

Zinc-Air batteries (which use oxygen from the air as one of the reactants) have the highest gravimetric and volumetric energy density of any battery chemistry, **6.0 x 10^9 J/m^3**. (1500 Wh/L) [11] Lithium-Ion batteries have the highest volumetric energy density of any stable rechargeable battery chemistry, **1.6 x 10^9 J/m^3**. (400 Wh/L) [10]

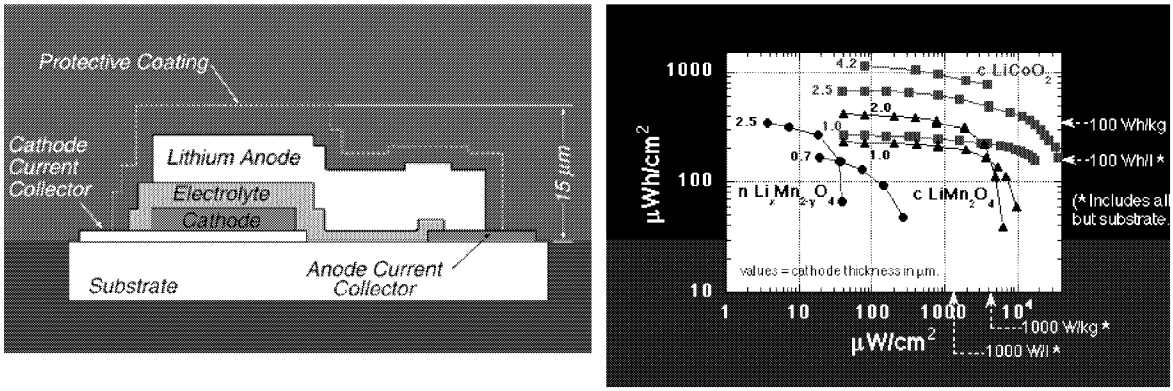

Figure 10. Microfabricated Rechargeable Batteries

A schematic drawing of the microfabricated battery described in [12] (left), and performance curves for thin-film rechargeable battery chemistries. (right) Both figures are from Oak Ridge National Laboratory.

By volume scaling, we can estimate that a 1 mm^3 zinc-air battery could about store **6.0 J**. This is enough energy to run a 2 μW sensing application for about 34 days, or a 10 mW heat-limited application for about 10 minutes.

Energy densities of microfabricated batteries from the literature support these estimates. In [12], workers at Oak Ridge National Laboratory report fabrication of a 1 cm^2 thin-film rechargeable Lithium battery, using photolithographic techniques, with an energy density of 2.1 x 10^9 J/m^3.

4.2.1.2 Microengines

Gasoline has an energy density of **2.9 x 10^{10} J/m^3**, making it one of the densest non-nuclear energy storage options available. In macroscopic devices, the energy in gasoline is converted to electricity by combustion.

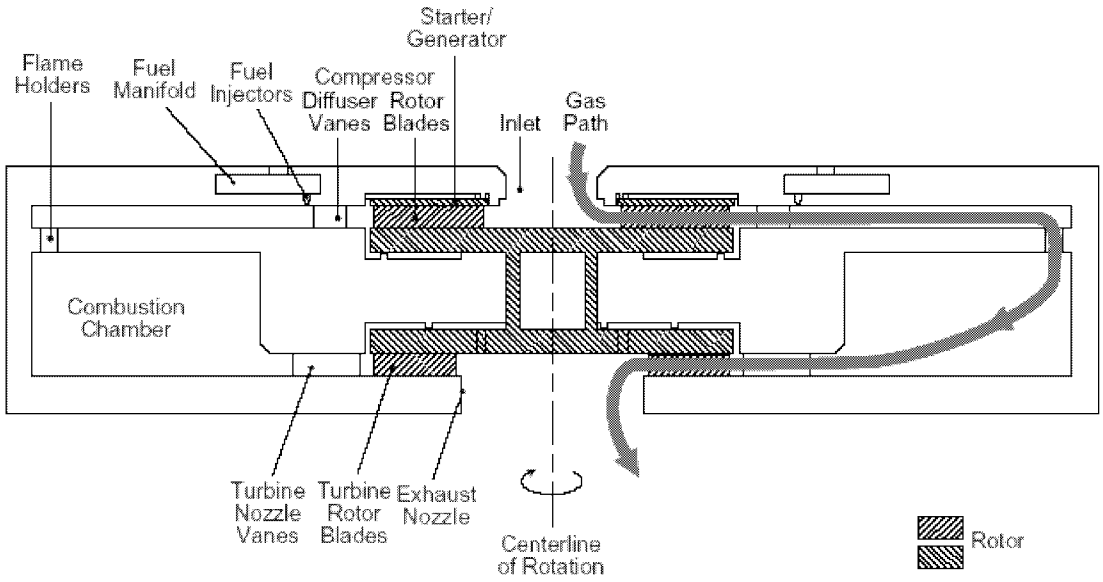

Figure 11. Micro Gas Turbine Generator Cross-Section (from [13])

The fabrication of millimeter-scale combustion engines is a topic of active research. To cite one example, the MIT Microengine Project [13] has built several prototype silicon gas turbine generators, which measure 4 mm on a side. The authors of [13] estimate that their engine will produce 10-20 W of electrical power while consuming 10 g/hr of hydrogen. By our calculation, this corresponds to a projected efficiency of 2.5% - 5%.

Using the above numbers, we can make a rough estimate of the effective electrical energy density of gasoline, **1.5 x 10^9 J/m^3**. This is about the same as the current energy density of microfabricated lithium-ion batteries. However, lithium-ion batteries are a very mature technology, while microengines are very new; it is likely that the efficiency of microengines will increase, perhaps dramatically, with further development.

A system powered by a microengine will be able to operate at much higher power levels than a battery powered system, due to forced convection of the exhaust. However, the presence of an exhaust stream from every particle in a paintable system might become irritating or dangerous to the operator, depending on the fuel selected.

We can estimate that 1 mm^3 of gasoline plus a microengine can store and convert **1.5 J**. This is enough energy to run a 2 μW sensing application for about 8 days, or a 10 mW application for about 2.5 minutes.

4.2.1.3 Radioisotopes

Radioisotopes can have tremendous volumetric energy density. ^{180}Ta, which has a half-life of eight hours, has an energy density of **1.0×10^{15} J/m^3**, about 30,000 times the energy density of gasoline. A 1 mm^3 sample of ^{180}Ta would continuously release 34 W. ^{178}Hf, which has a half-life of 31 years, has an energy density of **1.0×10^{16} J/m^3**, about 300,000 times that of gasoline. A 1 mm^3 sample would release 160 mW. [14]

However, two major factors currently restrict the usefulness of radioisotope power sources in paint particles: heat production and toxicity to humans.

Radioisotope power may be appropriate for systems that:

A. Require a very small amount of power
B. Must operate continuously for a very long time (e.g. 10-100 years)
C. Cannot be accidentally ingested or inhaled
D. Can be thoroughly gathered up and properly disposed of at end-of-life

4.2.1.3.1.1 Heat Production

To understand the heat production issue, suppose that a paint particle were powered by a 1 mm^3 sample of ^{178}Hf, and that all of the energy released by that sample was eventually converted to heat inside the particle. A 1 mm^3 particle has a 10 mW heat dissipation limit by natural convection, (see Figure 3) so the 160 mW emitted by the sample of ^{178}Hf would cause the particle to melt before leaving the factory.

To avoid this problem, a small enough volume of ^{178}Hf would need to be designed into the particle so that at most 10 mW of power was being released. However, microscale radioisotope energy conversion, though betavoltaic devices, tends to have a very low efficiency, about 1%. [15] With 1% conversion efficiency on the 10mW released, only 100μW of electrical energy would be available for use by the particle.

Unless the efficiency of microscale radioisotope energy conversion can be increased very substantially, or unless significant active cooling is used, radioisotopes cannot provide high power densities to paint particles, even before considering toxicity concerns.

However, radioisotope sources are capable of very high energy density, for very long life, very low power applications.

To continue our comparison from the previous section, *based on heat considerations alone*, a radioisotope source could power a 2μW sensing application for many years, but could not power a 10 mW application at all.

4.2.1.3.1.2 Toxicity

The main radioactive isotopes under active investigation for microscale power, ^{63}Ni and ^{3}H, are beta-particle emitters. Beta particles are electrons. The beta particles from these two isotopes have a low enough energy that they do not penetrate the outer layer of dead skin, or travel through more than a few inches of air. [16, 17]

However, if ingested or inhaled, beta particle emitters can still be highly toxic to humans, and can cause genetic damage, cancer, radiation sickness, and death. In many applications, including most of those listed in the introduction, there may be the potential for paint, paint particles, or materials that come into contact with paint particles to be accidentally ingested or inhaled.

The primary vector for radiation dose from ^{63}Ni is uptake into the bone, by inhalation of vapors or direct ingestion. [16] Ingestion of 20μCi/year of ^{63}Ni causes a radiation dose to the bone of 0.01 REM/year.

The U.S. Nuclear Regulatory Commission sets the dose limit to any part of the body to 0.1 REM/year for the general public, and 10 REM over five years for specially licensed *nuclear energy workers*. The occupational dose limit is liberal; if routinely exposed to the maximum dose over the course of a career, one would expect 25% of nuclear energy workers to die of cancer, versus 20% of the general population. [18, 19]

Scaling the ^{63}Ni dosimetry information, we can see that the NRC exposure limit for a member of the general public allows the ingestion or inhalation of a maximum of 200 μCi/year of ^{63}Ni, assuming that this was this person's only exposure to man-made radiation.

A prototype betavoltaic cell has been constructed using a ^{63}Ni source. [20] Extrapolating from experiments with lower-activity samples, the authors of [20] report that it should be possible to produce 150 nW of continuous electrical power from a 100 mCi source of ^{63}Ni. This is a very radioactive source; if a person were to accidentally ingest it, they could receive a radiation dose *500 times larger* than the annual dose limit for the general public, and *25 times larger* than the annual occupational dose limit.

Still, microscale betavoltaic batteries are a very new technology, and it is possible that they will improve in efficiency, perhaps by orders of magnitude, with further development. Also, it may be possible to encapsulate the radioactive material inside a

very durable protective coating, so that a person would not absorb the radioactive material into their body, even if they did ingest a paint particle.

4.2.2 Binder-Stored Energy

Binder-stored energy solutions store reactants or fuel in the paint binder, around the particles, rather than inside the particles. This allows relatively high-energy, long-lifetime, low-cost systems, compared to systems with particle-stored power.

Some materials development will be required to implement binder-stored energy in paintable systems; we know of no prototypes or demonstrations of this concept to date.

4.2.2.1 Inside-Out Zinc-Air Battery

An inside-out zinc-air battery, for a spray-on display application, is shown in Figure 12.

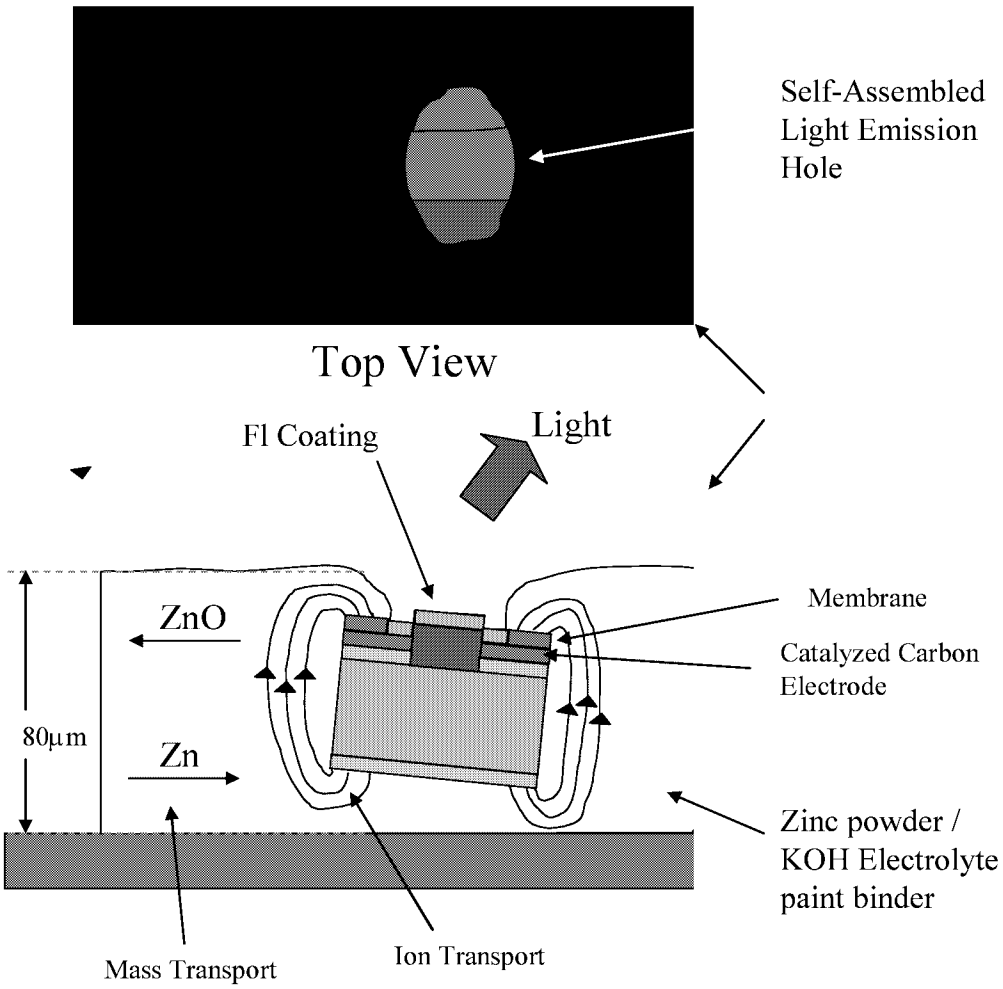

Figure 12. Inside-Out Zinc-Air Battery Concept for a Paintable Display Application
Paint particles consisting of a CMOS chip, LED, and battery electrodes are painted onto a surface in a Zinc powder / KOH electrolyte binder. The paint binder also contains polymers which harden into a porous matrix. Diffusion brings zinc from the binder to the particle, and takes the reaction product, Zinc Oxide, from the electrodes back into the binder.

With the inside-out zinc-air battery concept, the battery reactants are stored in the paint binder, rather than in the particles. Because of comparatively large volume of the binder, this results in longer battery life.

Consider a paintable display with a paint thickness of 1 mm, and with cubic 110 μm particles, each drawing 50μW, with one particle for every 0.25 mm^2 area. (These are the specifications for the 640 x 480, 17" paintable display evaluated at the end of this study.)

If particle-stored power were used, then a maximum of 0.0013 mm^3 would be available for energy storage inside each particle. This would be enough volume to store 8 mJ of energy using a zinc-air battery, so the battery life of this display would be limited to 2.5 minutes.

With the inside-out zinc air battery, an example of binder-stored power, about 0.25 mm^3 per particle is available for the storage of energy, which is enough volume to store up to 1.5 J per particle. In this case, the battery life of the display could be up to 8 hours.

The zinc and zinc oxide must travel through the polymer matrix in the binder by diffusion. We can get a rough idea of the particle size required to achieve suitable power density by evaluating the diffusion time constant for a zinc particle. The diffusion time constant should be about equal to the desired service lifetime of the battery.

$$\tau_D = \frac{d^2}{\pi^2 D}$$ *Equation 1. Diffusion Time Constant*

$$D = \frac{kT}{6\pi\eta\sigma}$$ *Equation 2. Stokes-Einstein Relation*

$$\sigma = \frac{\pi}{6} kT \frac{\tau}{\eta d^2}$$ *Equation 3. Combining Equations 1 & 2*

In Equations 1-3, τ_D is the diffusion time constant, d is the length scale for diffusion, D is the diffusivity of the zinc particles, k is Boltzmann's constant, T is the absolute temperature, η is the dynamic viscosity of the liquid phase of the binder, and σ is the zinc particle size. Taking T = 300 K, τ_D = 8 hours, η = 8.7 x 10^{-4} kg/m-sec, the viscosity of water, d = 1mm, and k = 1.38 x 10^{-23} m^2-kg/s^2-K, then σ, the maximum zinc particle radius, is 72 nm. It is possible to fabricate zinc nano-flakes as small as 3-5 nm in diameter by dry roller vibration milling [21], so it is possible to fabricate zinc particles small enough for the inside-out zinc-air battery discussed here.

4.2.2.2 Binder-Stored Fuel

In the combustion engine with binder-stored fuel concept, each particle contains a combustion engine which is powered by fuel drawn in by capillary action or diffusion from pores in the paint binder, and by oxygen from the air. Similar energy densities to the inside-out zinc-air battery are possible, with potentially larger power densities.

4.2.3 External Power

In this section we consider technologies for supplying power to a paintable system from an external source; this power might come from the commercial power grid or from a vehicle's electrical system.

In §4.2.1-§4.2.2, we have shown that it is possible to run paintable systems, even power hungry ones, on battery power for several hours. This is on par with the battery life of ordinary macroscopic systems, like laptop computers and cellular telephones. When the batteries in these devices run out, they can be plugged in and recharged. This section is about how to "plug in" a paintable system, either to recharge its batteries, or to allow for continuous operation.

Acceptable methods for supplying power to a paintable system cannot require hand-manipulation of individual particles. Having personnel attach tiny connectors to each of the millions of particles that make up a paintable system would be absurd. Also, power transfer through conductive planes is not ideal, because a single shorted particle could short out the entire system, and because the particles would need to be permanently attached to the planes. What is needed is a hands-off, wireless, batch process for supplying power the sparsely distributed, randomly oriented paint particles.

There are a wide range of possible options; for a general survey of power harvesting techniques for mobile electronics, see [22]. Here we focus on two options that we feel are the most feasible for paintable systems: photovoltaic cells and reactive power transfer.

4.2.3.1 Photovoltaic Cells

Particles powered by photovoltaic cells can operate indefinitely in a random environment, so long as it is not too dark. This is a claim that cannot be made regarding any of the other power sources discussed in this document, with the possible exception of the radioisotope sources.

Since high-efficiency photovoltaic cells are already made using wafer-fabrication technology, little technology development would be required to use them as an energy source for paint particles.

The full-sun outdoor solar irradiance is 100 mW/cm^2. Indoor irradiance is typically less than 1 mW/cm^2. [23] The best single crystal cells, fabricated using a GaInP / GaAs process, have 30.3% efficiency. Single-crystal silicon cells have an efficiency of 24.7%. [24]

Applying these figures to a 1 mm^2 cell, photovoltaic cells can deliver **300 μW** outdoors, when the sun is shining, and **3.0 μW** indoors, when the lights are on.

Shining a lamp on sparely-distributed paint particles equipped with photovoltaic cells, in an attempt to transfer power, results in poor efficiency. The most efficient lamps available are sodium-vapor lamps, which have an efficiency of about 34%. We can multiply this by the efficiency of the GaInP / GaAs solar cells to get a direct transfer efficiency of 10%. But then, we need to multiply this efficiency by the area-fill-factor of the particles on the painted surface to get the overall efficiency, since light that does not hit a particle is wasted. The paintable display application discussed here has an area fill factor of 5%, which results in an overall power transfer efficiency of 0.5%. For an application with higher fill factor, this approach might be acceptable.

4.2.3.2 Reactive Power Transfer

High efficiency wireless power transfer to sparely distributed, randomly oriented paint particles on a surface can be realized by placing the paintable system inside the inductor or capacitor of a resonant LC tank circuit. This approach is highly efficient because energy that is not absorbed by a particle during one AC cycle is not wasted; most of it is recaptured and used again on the next cycle.

To test this concept, we constructed a prototype paintable system; using 0603 LED's as paint particles. An 0603 LED is about 1 mm long. The prototype system supplied about 120 μW to each of the LED's.

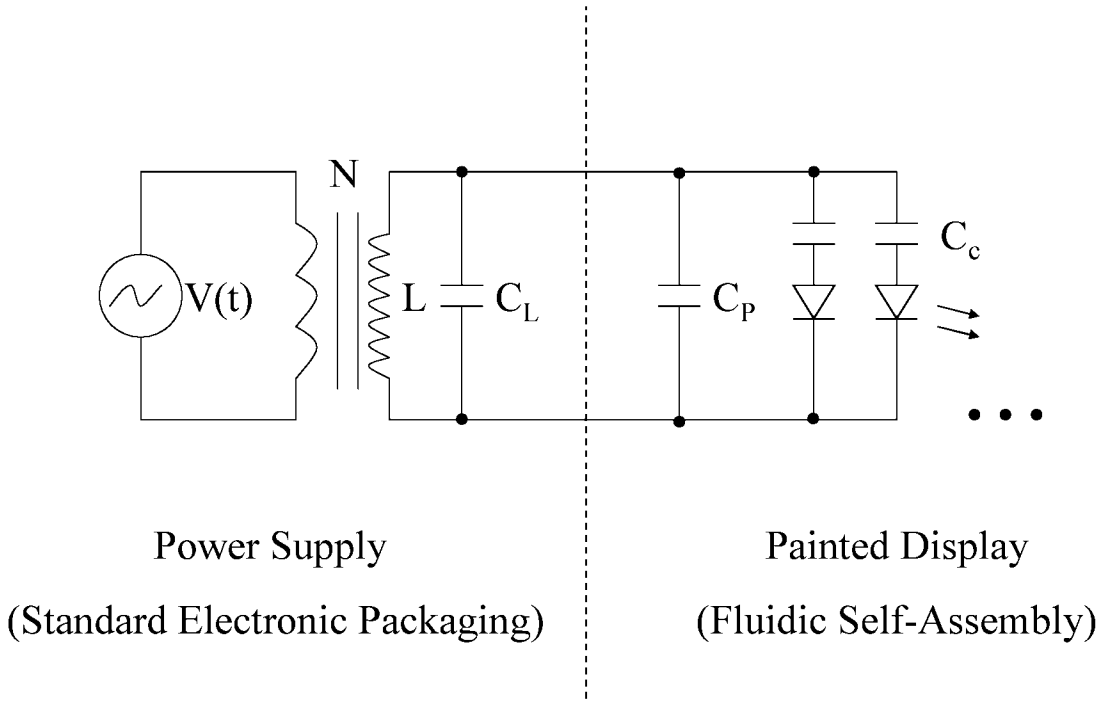

Figure 13. Electrical Model for Reactively Powered Paint

An AC voltage source energizes a transformer-coupled parallel LC tank circuit. The paint nodes (shown as LED's) are placed inside the capacitor, which results in their being capacitively coupled to the parallel LC network.

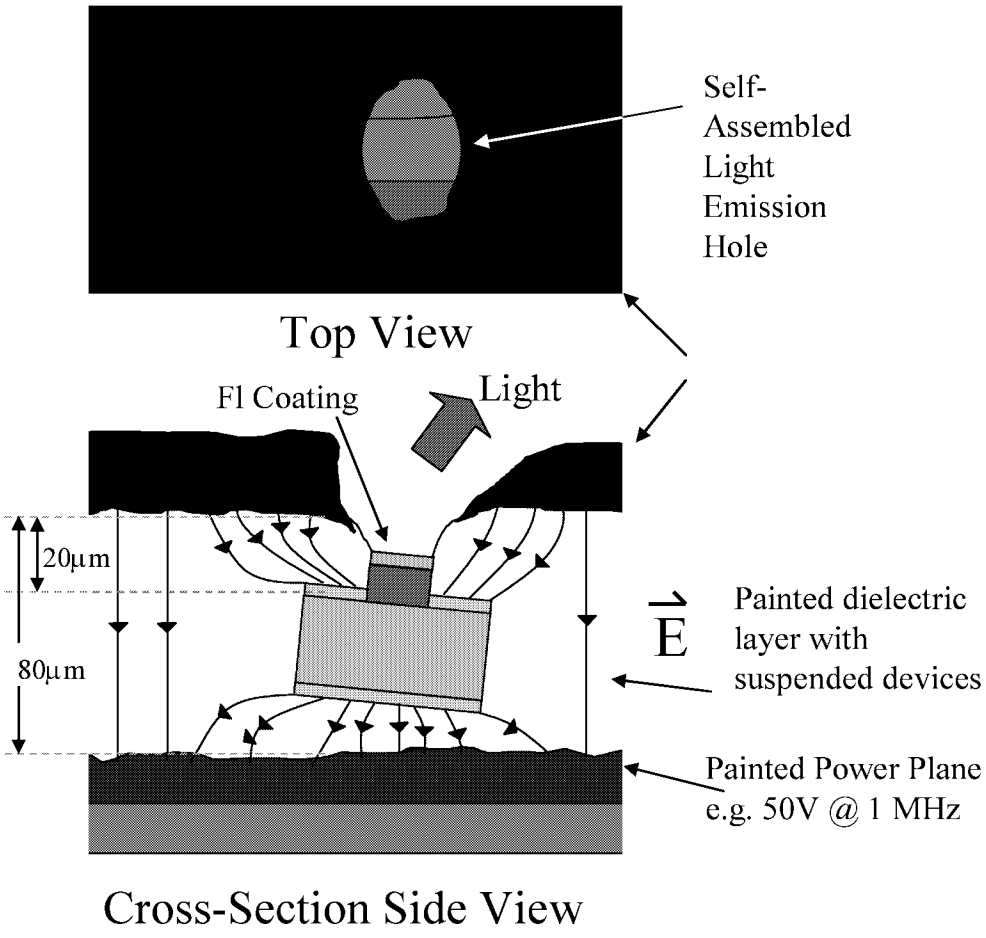

Figure 14. Reactive Power Transfer Concept for a Paintable Display

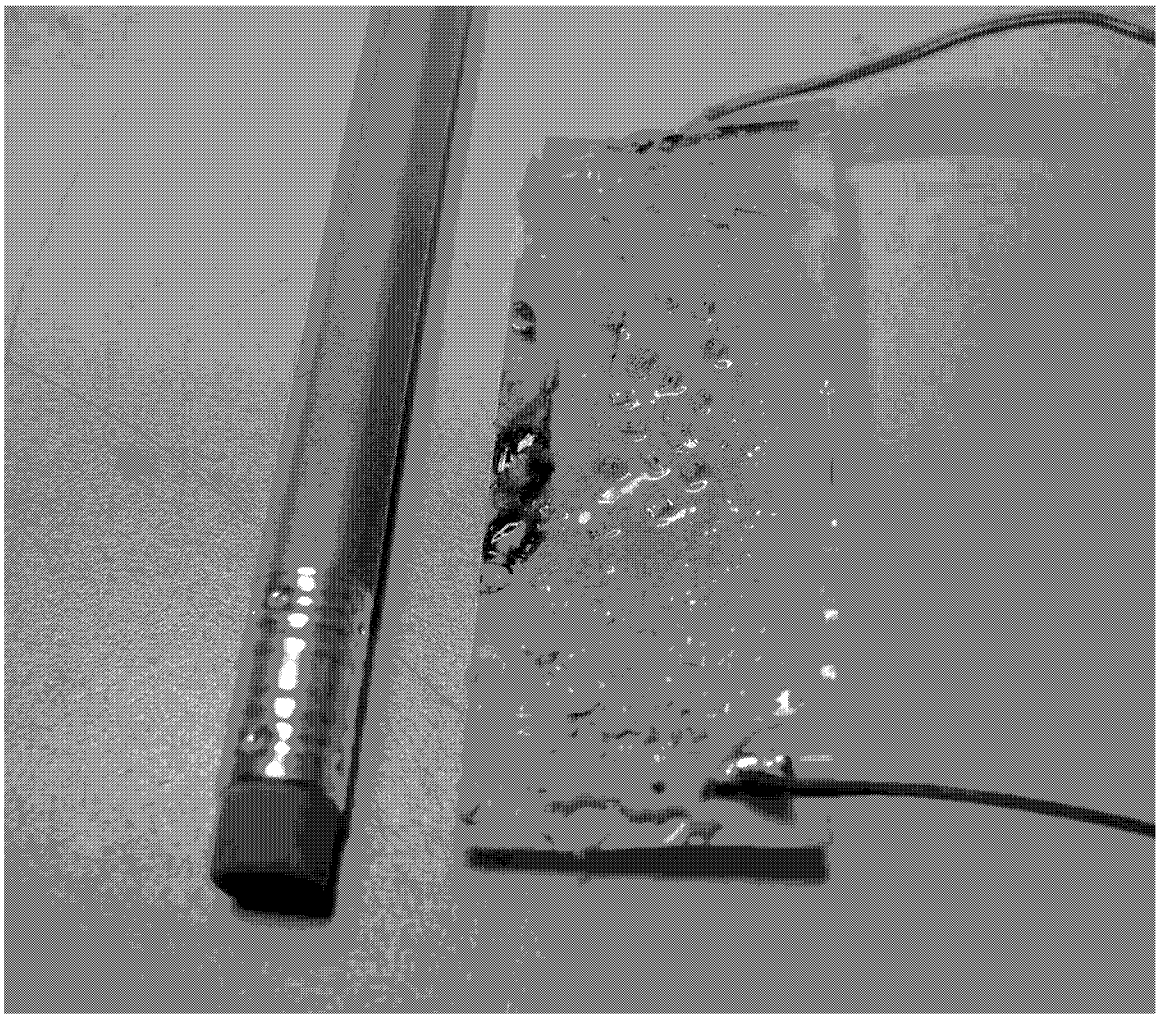

Figure 15. Reactive Power Transfer Demonstration

We fabricated a single-layer flex circuit with interdigitated electrodes. We coated the circuit with a dielectric paste, made by loading a two-part urethane epoxy to the thickening point with nanophase barium titanate. The paste has a relative dielectric constant of about 9.0. We placed 0603 LED's on top of the paste in random positions and orientations. (The brown paste in the center is from an earlier, non-functional formulation.) We excited the electrodes with 400 VRMS at 500 kHz; two illuminated LED's are clearly visible.

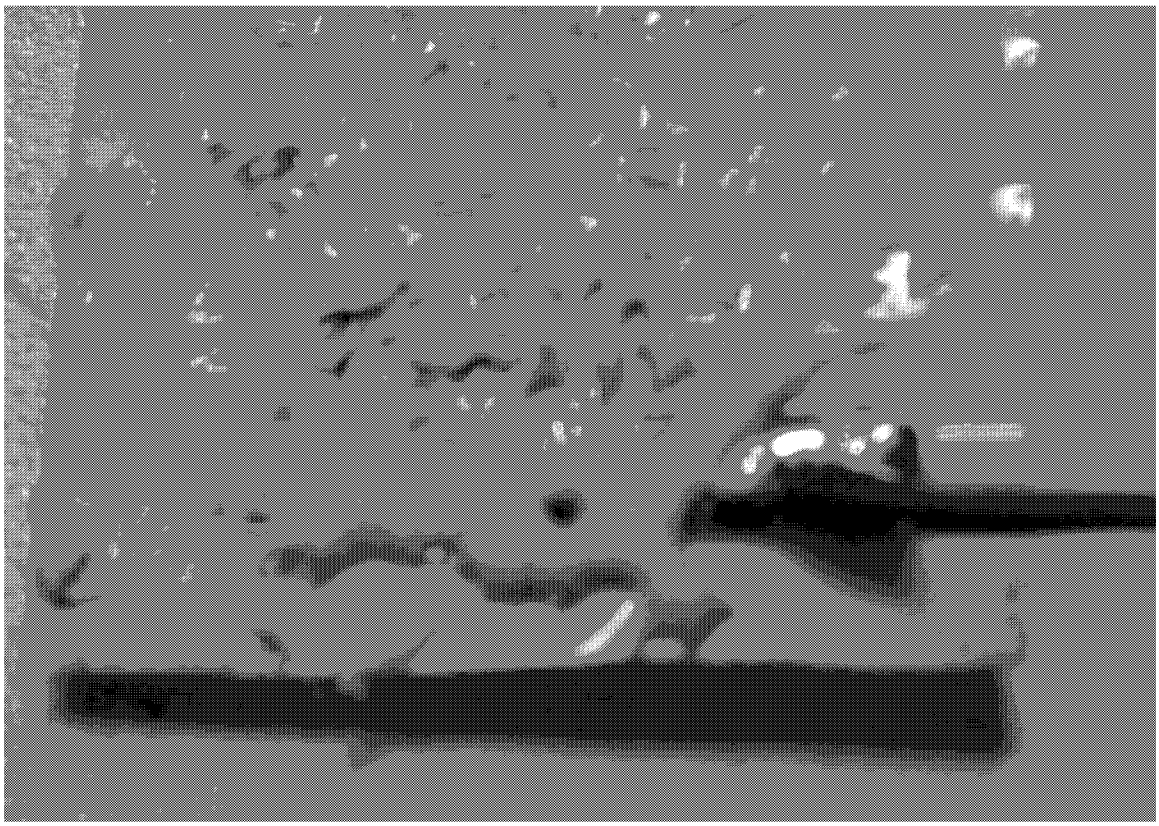

Figure 16. Reactive Power Transfer Demonstration: Close Up

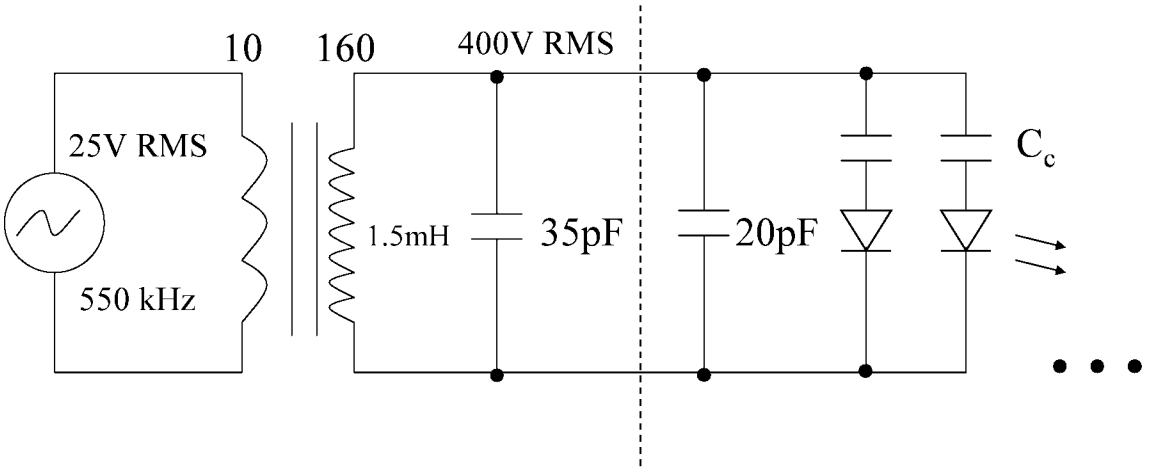

Figure 17. Electrical Details of Reactive Power Transfer Demonstration

A 550 kHz sine wave, produced by a function generator (HP 33120A) is amplified to 25 VRMS by a power op-amp (Apex PA09) and applied to the primary of the transformer. The secondary of the transformer, at 400 VRMS, is connected to a 35 pF capacitor, used for tuning, and to the flex circuit containing the LED's. We measured the capacitance of the flex circuit at 20 pF. From geometry, we estimate the coupling capacitance between the LED pads and flex circuit at 62 fF, which would supply 120 μW to each LED. This value is commensurate with the level of illumination observed.

It is also possible to place the paint particles inside the inductor of an LC tank circuit. In this case, it would be desirable for the paint binder to be made from a ferromagnetic material. Transparent high permeability materials and polymer composites of these materials exist; see [25]. Also, it would be desirable to place even more strongly ferromagnetic materials inside each particle; for an example of ferromagnetic materials integrated with a wafer-fabrication process, see [26].

4.2.3.3 Power and Efficiency Calculations

From Figure 13, the electrical model for capacitive power transfer is a load in series with a small capacitance for each particle, plus a large parallel shunt capacitance. In this section, we establish approximate mathematical expressions for the maximum achievable power transfer and efficiency to each particle.

The power into a single particle is given by

$$P_p = \frac{V_p^2}{R_p}$$ *Equation 4. Power to a particle*

where P_p is the power transfer to the particle, V_p is the voltage across the particle's terminals, and R_p is the resistance between the particle's terminals. Maximum power transfer from a source to a load occurs when the source impedance equals the load impedance; in this case

$$R_p = \frac{1}{2\pi f C_s}$$ *Equation 5. Impedance Matching Condition*

where f is the operating frequency and C_s is the series capacitance to the particle. We can combine Equations 4 & 5 to get the maximum power into a particle.

$$\left(P_p\right)_{\max} = 2\pi V_p^2 f C_s$$ *Equation 6. Combining Equations 4 &5*

We can estimate the series capacitance as

$$C_s = \varepsilon_o \kappa \frac{A_p}{(L_b - L_p)}$$ *Equation 7. Series Capacitance*

where ε_o is the permittivity of free space, κ is the relative dielectric constant of the paint binder, A_p is the area of each particle electrode, L_b is the thickness of the paint layer, and L_p is the thickness of the particle.

Combining Equations 6 & 7, we get an expression for the maximum power into a particle in terms of material properties and geometry.

$$\left(P_p\right)_{\max} = 2\pi V_p^2 f \varepsilon_o \kappa \frac{A_p}{(L_b - L_p)}$$ *Equation 8. Combining Equations 6 & 7*

Now we make a gross approximation. We estimate that the thickness of the painted film is controlled to within a tolerance of 10%. This means that, if we try to make the film thickness equal to the particle thickness, to try to get maximum power transfer, then in the worst case,

$$L_b - L_p = \frac{1}{10} L_p$$ *Equation 9. Paint Dimensional Tolerance*

We can now combine Equations 8 & 9 to get a simple estimate of the power transfer efficiency to a paint particle by capacitive means.

$$\left(P_p\right)_{\max} = (20\pi) \cdot (\varepsilon_o \kappa) \cdot V_p^2 \cdot f \cdot L_p$$ *Equation 10. Capacitive Power Transfer*

Note the linear scaling of power with particle dimension. As the length scale gets smaller, the length to volume ratio goes up. This means that capacitive power transfer becomes capable of supplying a larger amount of power, relative to photovoltaic cells or internal energy storage, as the particle size considered becomes smaller.

For a 1 mm particle, a paint binder with a relative dielectric constant of 9.0, a resonant frequency of 1 MHz, and a semiconductor process with V_{DD} = 3.3 V, P_{max} = **100µW**. If we use a higher voltage process, such as HVCMOS, with V_{DD} = 35V, then P_{max} = **12 mW**. In this case, an on-particle switching converter [26] or linear regulator would be required to power digital circuitry at a lower voltage.

For our 110 µm paintable display particles, which consume 50µW, we would need f = 8 MHz to transfer the required power with V_{DD} = 3.3 V. Alternately, we could make V_{DD} = 10V (potentially by stacking some LED's in series) and use f = 1.0 MHz.

The efficiency of a power transfer system is defined to be

$$\eta = \frac{E_{delivered}}{E_{delivered} + E_{lost}}$$ *Equation 11. Definition of Efficiency*

where η is the efficiency, $E_{delivered}$ is the energy delivered to the load, and E_{lost} is the energy lost to the surroundings.

The energy lost per cycle in an RLC network is given by the Q of the network,

$$Q = \frac{E_{stored}}{E_{lost}}$$ *Equation 12. Definition of Q*

where E_{stored} is the energy stored per cycle.

Since the magnitude of the impedance of the series capacitance and the particle are equal, and since they are in series with each other, we have,

$$\left(E_{stored}\right)_{series} = E_{delivered} \qquad \textit{Equation 13. Matching Condition}$$

The energy stored in the parallel capacitance, covering the parts of the paintable system where there are no particles, is given by

$$\left(E_{stored}\right)_{parallel} = \frac{1}{2} C_{parallel} V^2 \qquad \textit{Equation 14. Energy Stored in a Capacitor}$$

where $C_{parallel}$ is the energy stored in the parallel capacitance per cycle, and V is the voltage across the capacitor plates. Since the impedance of the series capacitance and particle are matched,

$$V = 2V_p \qquad \textit{Equation 15. From Matching Condition}$$

We can write an expression for the parallel capacitance in terms of the paint area without a particle, A_b,

$$C_{parallel} = \varepsilon\kappa \frac{A_b}{L_b} \qquad \textit{Equation 16. Parallel-Plate Capacitor}$$

Assuming cubic particles with length L_p, that $L_b \approx L_p$, and given an area fill-factor of F for particles on the surface, we can write,

$$C_{parallel} = \varepsilon\kappa \frac{L_p^2\left(\frac{1}{F} - 1\right)}{L_b} \approx \varepsilon\kappa L_p\left(\frac{1}{F} - 1\right) \qquad \textit{Equation 17. Parallel capacitance}$$

We can find the energy stored in this capacitance by combining Equations 14 & 17,

$$\left(E_{stored}\right)_{parallel} = 2\varepsilon\kappa L_p V_p^2\left(\frac{1}{F} - 1\right) \qquad \textit{Equation 18. Energy in Parallel capacitance}$$

Modifying Equation 10, the energy delivered in a single cycle is

$$E_{delivered} = 20\pi\varepsilon_o \kappa L_p V_p^2$$ *Equation 19. Energy Delivered*

Combining Equations 13, 18, and 19, we can write the total energy stored per cycle.

$$E_{stored} = \varepsilon_o \kappa L_p V_p^2 \left(20\pi + \frac{2}{F} - 2 \right)$$ *Equation 20. Energy Stored*

Combining Equations19 & 20, we get the ratio of energy stored to energy delivered.

$$\frac{E_{stored}}{E_{delivered}} = \frac{20\pi + \frac{2}{F} - 2}{20\pi} = 1 + \frac{\left(\frac{1}{F} - 1 \right)}{10\pi}$$ *Equation 21. Energy Ratio I*

Combining Equations 12 & 21, we get the ratio of energy lost to energy delivered.

$$\frac{E_{lost}}{E_{delivered}} = \frac{E_{stored}}{E_{delivered}} \cdot \frac{E_{lost}}{E_{stored}} = \left[1 + \frac{\left(\frac{1}{F} - 1 \right)}{10\pi} \right] \frac{1}{Q}$$ *Equation 22. Energy Ratio II*

Combining Equations 11 & 22, at long last, we get an expression for the efficiency.

$$\eta = \frac{10\pi Q}{10\pi(1+Q) + \frac{1}{F} - 1}$$ *Equation 23. Capacitive Power Transfer Efficiency*

This is the expression for the efficiency of capacitive power transfer to paint particles, assuming a large, flat paint layer with a thickness tolerance of 10%, sandwiched inside of a large parallel-plate capacitor. F is the area fill factor of paint particles, and Q is the quality factor of the LC network, before the addition of particles.

We can estimate Q at 40; this is a reasonable guess. The paintable display application detailed at the end of this report has an area fill factor of about 5%. From Equation 23, assuming perfect impedance matching and operation at resonance, capacitive power transfer has an efficiency of **96%**, similar to a good switching power supply. To get this kind of efficiency in practice, a feedback control system would probably be needed to keep the system precisely at resonance over variations in load.

4.3 Communications

4.3.1 Introduction

The paintable computing architecture assumes local communication; each particle can communicate with about 10 other particles in its local neighborhood; typically over a distance of less than 1 cm. Messages intended for particles further away need to be forwarded by the network.

This architecture is scalable to networks with millions of nodes; an architecture in which any paint particle could communicate with any other via a high-power transmitter would not be scalable, and would have higher total power consumption. There are three main reasons why local communication is scalable.

First, it reduces operating system overhead. If a particle had thousands of nodes inside its communications radius, it would be constantly processing and discarding messages not intended for it, and this would use processing resources and power.

Second, it reduces clutter on the communications channel, increasing the total available bandwidth. With large communications radiuses, the channel would always be jammed up with communication between far-off particles, so the data rate between any two particles would slow to a crawl.

Third, because of the inverse-square attenuation of electromagnetic waves, the required total transmit power goes as the square of the distance between nodes for direct communication, but linearly with that distance for multihop communications. In the large distance limit, it takes lower total power to forward a message through a multihop network than to send it directly. [27]

In the following three sections, we discuss free-space optical, propagating-wave RF communications, and near-field RF communications.

Near-field RF communication has a very precise radius, and is not subject to multipath or fading, so it can be used for accurate particle localization. Optical communication has the lowest minimum die area requirement, and low peak power consumption. Optical communication is strictly line of sight. Both optical and near-field RF communications allow power efficiency as high as 30 pJ/bit.

Propagating-wave RF communication may be possible from volumes as small as 1 mm^3, due to recent advances in antenna design and circuit techniques at 60 GHz. Propagating wave communication would be most useful between paint nodes with a very low fill factor; for example, paint nodes separated from one another by several meters.

There has been a recent revolution in RF CMOS circuit techniques. Except for very high performance systems, III-V is no longer required for RF. [28, 29, 30]

4.3.2 Communications Transports

4.3.2.1 Optical

Optical emitters and detectors (IR) can fabricated on silicon, and the peak receiver power level required is very low. Optical emitters are directional. Diffusing lenses or a translucent, milky paint binder would be needed to insure communication between neighbors.

The intensity of an optical signal drops off with the inverse square of distance. The intensity of the emitter can be controlled by current regulation, starting low and ramping up, to establish a properly-sized neighborhood.

The circuitry and detectors for optical communications systems can be very small, and typically do not require external passive components. Workers at U.C. Berkeley built a "smart pixel" integrated optical receiver inside a 150 μm square, using a 350 nm process. The data rate was 2.5 Mb/sec, and the receiver used only 50 μW to achieve a -51dbm sensitivity. [31]

To produce -51 dBm at 10 mm from a 1 mm source requires -31 dBm of optical power. Given the ~10% efficiency of LED's, this requires -21 dBm of electrical power, about 8μW.

Using the figures listed above, for 1 mm nodes and 10 mm neighborhoods, we can estimate the total power consumption for the communications system at about 60 μW. This corresponds to an efficiency of **24 pJ/bit.**

4.3.2.1.1 Near-Field RF Communication

When particles are spaced by less than a few wavelengths, near-field communication becomes possible. Near-field communication works by capacitive or inductive coupling, rather than by propagation of electromagnetic waves.

Coils or capacitor plates for near-field communication can be much smaller than antennas at a given frequency: this allows the use of lower-frequency bands and less exotic circuit techniques than with propagating-wave communication.

Power drops off with the sixth power of distance. This gives brick-wall neighborhoods, makes eavesdropping and jamming all but impossible, and results in very low interference levels. However, it also restricts this technique to systems where the neighborhood size is physically small.

Nearfield systems are preferable for localization based on received signal strength, because, unlike propagating-wave systems, they are not subject to multipath or fading. In

a near-field system, the received signal strength is a function only of the transmit power, the separation distance, and the presence of any dielectric (in the case of electric-field systems) or magnetic (in the case of magnetic-field systems) materials inside the neighborhood. Because magnetic objects (e.g. steel bolts) are rarer in nature than dielectric objects (e.g. raindrops, insects, people), magnetic-field systems are preferable for precise localization.

4.3.2.2 Near-Field Inductive Communications System Design

In this section, we present a block diagram and performance calculations for an inductive communications system. This system is designed for 1 mm particles and a 10 mm neighborhood size.

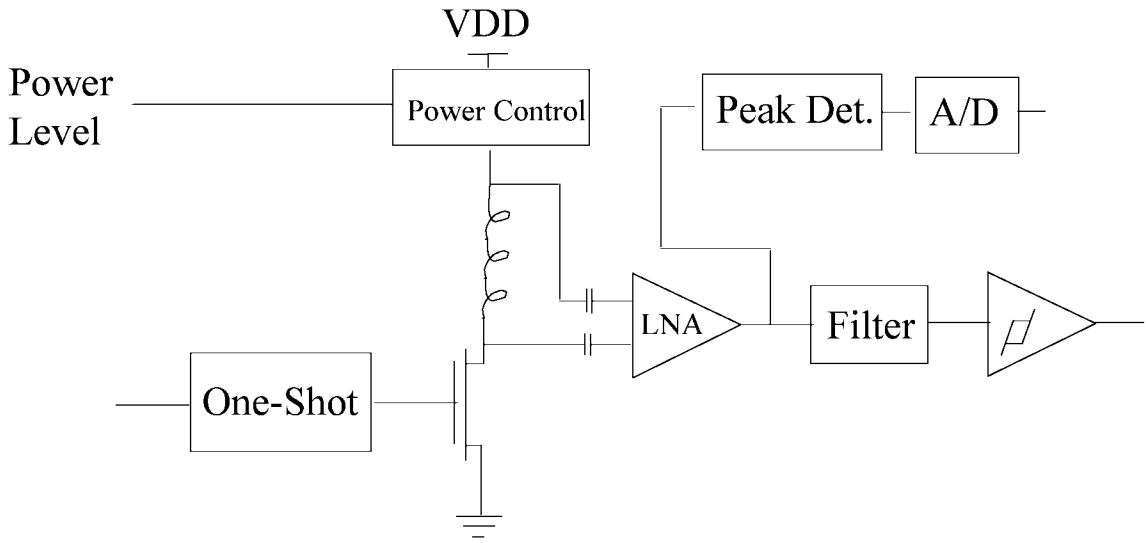

Figure 18. Block Diagram: Inductive Communications System

This system includes power control and RSSI, for precise control of neighborhood size, precise localization, and very high data rate.

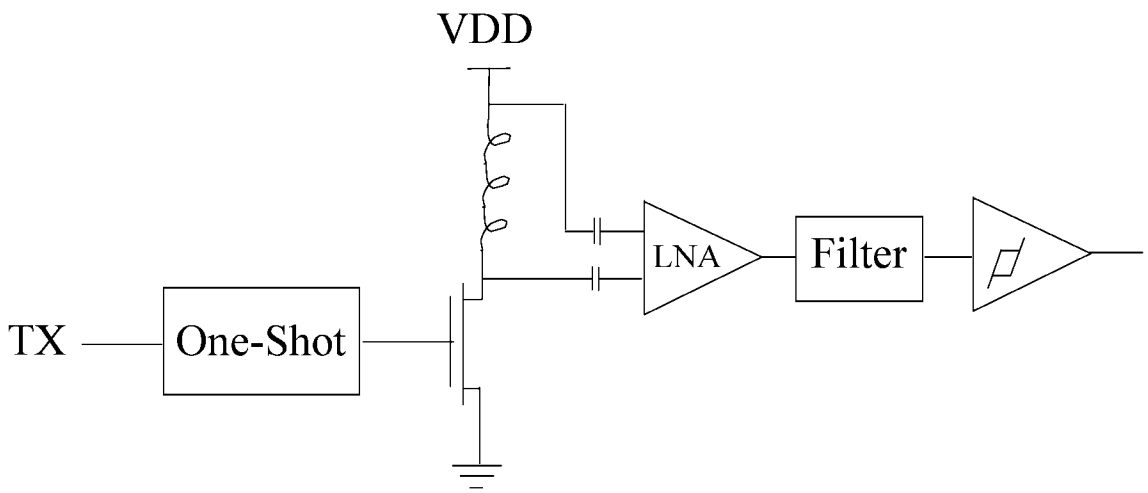

Figure 19. Block Diagram: Inductive Communications System

This system aims for minimum transistor count. All of the blocks shown are simple analog elements and require just a few transistors each; the total transistor count for this circuit is probably 20-30.

In this section, we will calculate the resistance, self-inductance, and mutual inductance of the inductors that couple data between particles. We will use these figures to calculate the path loss at 10 mm, signal to noise ratio, bandwidth, transmitter and receiver power consumption, and maximum data rate.

For this design example, we select an Intel 130 nm CMOS process, and a die size of $1mm^2$. We place the communications inductor on metal layer 6 (see Figure 20 for design rules) and select $d_{in} = 0.8$ mm and $d_{out} = 1$ mm. (See Figure 21) We select an analysis frequency of 1 GHz, corresponding to the transmission of 1 ns pulses.

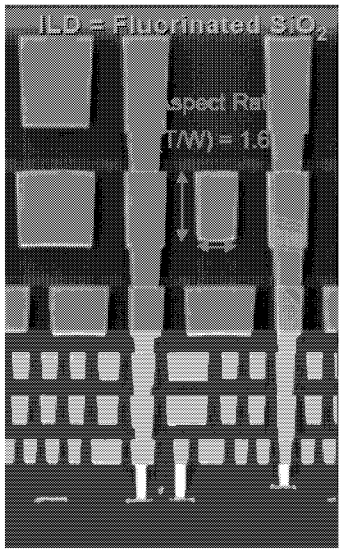

Figure 20. Scanning Electron Micrograph of a 130nm CMOS IC, Cross Section

The metallization pitch is 350 nm for the bottom metal layer and 1200 nm for the top metal layer, number 6. Upper layer traces are 600 nm wide by 960 nm high. From [40].

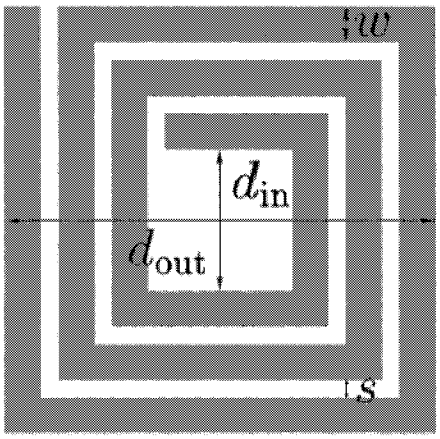

Figure 21. Planar Integrated Circuit Inductor from [32]

First, we need to compute the diameter and area fill factor of the inductor.

$$d_{avg} = \frac{d_{in} + d_{out}}{2}$$ *Equation 24. Average Diameter (from [32])*

$$\rho_{fill} = \frac{d_{out} - d_{in}}{d_{out} + d_{in}}$$ *Equation 25. Inductor Fill Factor (from [32])*

For this design, using the above expressions, d_{avg} = 0.9 mm and ρ_{fill} = 0.1111. Given small ρ_{fill} and thin traces, so that the skin effect is not an issue, we can write

$$R = \rho \frac{L}{A} \cong \rho \frac{4 d_{avg} N}{W_{turn} H_{turn}}$$ *Equation 26. Coil Resistance*

In this equation, ρ is the resistivity of the inductor trace, L is the total trace length, A is the cross-sectional area of the trace, N is the number of turns, W_{turn} is the width of the trace, and H_{turn} is the height of the trace.

We select W_{turn} = 8 μm and set H_{turn} = 960 nm, corresponding to the value given in Figure 20. Given these trace dimensions, we can fit about 6 turns, so N = 6. The traces are made of copper, so we set $\rho = 1.6 \times 10^{-8}$ Ω/m.

From Equation 26, we get R = 49.2 Ω. This is not an accident; we selected the trace width to make the resistive part of the impedance of the inductor as close as possible to the standard 50 Ω.

A standard expression for the inductance of a planar spiral is,

$$L = 2.34 \mu_o \frac{N^2 d_{avg}}{1 + 2.75 \rho_{fill}}$$ *Equation 27. Self-Inductance (from [32])*

where μ_o is the magnetic permeability of free space, equal to 1.26 x 10^{-6} H/m. From this expression, we see that the inductance of our inductor is 79 nH.

We can calculate the reactive part of the impedance of the inductor using

$$|X_L| = 2\pi fL \qquad \textit{Equation 28. Reactive impedance}$$

The reactive part of the impedance of the inductor at 1 GHz is 497 Ω. This is good news: it means the inductor looks 10 times more like an inductor than a resistor at 1 GHz.

The mutual inductance between two identical inductors oriented in the same direction is given by

$$L_M = \frac{\mu_o \pi N^2 (d_{avg}/2)^4}{2r^3} \qquad \textit{Equation 29. Mutual Inductance, from [33]}$$

where L_M is the mutual inductance and r is the separation distance between the inductors. Taking r = 10 mm, equal to the neighborhood size, L_M = 3.2 pH.

We can compute the induced voltage by combining the expressions for self and mutual inductance. When the system transmits a pulse, V_{DD} is forced across the transmitting inductor. This induces a current in the transmitting inductor according to

$$V_{DD} = L\frac{dI_T}{dt} \qquad \textit{Equation 30. Self Inductance}$$

where I_T is the current in the transmit inductor and t is time. This time varying current produces a time-varying magnetic field, which induces a voltage in the receive inductor according to

$$V_r = L_M \frac{dI_T}{dt} \qquad \textit{Equation 31. Mutual Inductance}$$

where V_r is the voltage induced in the receive inductor. Combining Equations 30 & 31, we get an expression for the voltage of the received signal in terms of V_{DD},

$$V_r = \frac{L_M}{L} V_{DD} \qquad \textit{Equation 32. Induced Voltage}$$

For a design using the Intel 130 nm process, VDD = 1.2V, so V_r = 47 μV. This voltage appears in series with the 50 Ω resistive impedance of the inductor, so we can compute the received power using

$$P_{signal}(dbm) = 10\log_{10}\left(\frac{V_r^2}{(50\Omega)(1mW)}\right) \qquad \textit{Equation 33. Signal Power at the Receiver}$$

where $P_{signal}(db)$ is the signal power at the receiver, in decibels relative to 1 mW. Using this expression, we calculate that the signal power at the receiver is -73 dBm.

We can calculate the noise power using

$$P_{noise}(dbm) = 10\log_{10}\left(\frac{kTf}{Q(1mW)}\right) + NF(db) \qquad \textit{Equation 34. Noise Power}$$

where $P_{noise}(dbm)$ is the input-referred noise power at the receiver, Q is the quality factor of the receive amplifier, related to the half-power bandwidth, and NF(db) is the noise figure of the receive amplifier in decibels.

Selecting a receive amplifier with Q = 10 (for a bandwidth of 100 MHz at f = 1 GHz), and a noise figure of 3 dB, both of which are readily achievable using an inductively-degenerated CMOS LNA [30], we can use Equation 34 to calculate that the input-referred noise at the receiver is about -90 dbm. This gives a signal-to-noise ratio of 17 dB, about 50 expressed as a straight power ratio.

We can estimate the communications bandwidth for the channel using the Shannon capacity. At one time, this would have been a gross over-estimation, but modern coding techniques (e.g. Turbo codes) can come very close to the Shannon capacity.

$$C = W\log_2(1+SNR) \qquad \textit{Equation 35. Shannon Capacity}$$

Substituting W = 100 MHz and SNR = 50, we get a channel capacity of 567 Mb/s.

We can get bitrates close to the channel capacity using 8-QAM modulation and turbo coding. Alternatively, we can use OOK modulation, as is shown in the block diagrams, and accept lower data rate, probably around 50 Mb/s, in exchange for reduced area.

Based on designs given in [30], we can guess that a 1 GHz, Q = 10 inductively degenerated LNA with 3 dB NF will require about 10 mW. This is likely to be the dominant source of receiver power; we estimate the receiver power at 1.5 times this figure, which is 15 mW.

We can estimate the transmitter power using

$$P_T = 2\frac{V^2}{|R + jX_L|} \qquad \textit{Equation 36. Transmitter Power Estimate}$$

which assumes that the power stored in the inductor is completely dissipated on every cycle, and Class-A (worst case) amplification. It may be possible to do much better than this with Class-D amplification and power recovery from the inductor. From Equation 36, we can estimate the transmitter power at 6 mW.

$$\left(\frac{E}{bit}\right)=\frac{P_D}{C}$$ *Equation 37. Energy per Bit*

Combining all of our assumptions and approximations, some liberal and some conservative, we can power efficiency of this communications system, to a node at the edge of the neighborhood, at **37 pJ/bit**.

4.3.2.3 Propagating-Wave RF Communication

The most obvious challenge to building a propagating-wave RF transceiver inside of 1 mm^3 is the size of the antenna.

Commercial RF systems use external antennas, so the transmitter output and receiver input are almost always matched to the industry-standard 50 Ω impedance. The radiation resistance of an antenna is related to its length; for example a 75 Ω antenna must have an electrical length of about 1/2 wavelength. [34] A 1-mm long, 1/2 wavelength antenna would have a center frequency of 150 GHz. The design of electrically shorter antennas and the corresponding low impedance RF circuits to drive them is possible in principle. Of course, the design of higher frequency RF circuits, still operating at the same comfortable impedance, is also possible in principle.

There is a recent body of work concerning techniques for fabricating integrated antennas. In one result, workers fabricated a 2-mm long zigzag dipole antenna, and observed proper inverse-square dependence over 4-5 meters, at a frequency of 24 GHz. [35]
A great deal of work is ongoing to develop circuit techniques for low-power RF communications. For example, workers fabricated a complete 433 MHz UHF radio transceiver, with 24 kb/s data rate and 1 mW power consumption. [36]

The FCC recently created a new ISM band at 60 GHz, with 7 GHz bandwidth. As a result, the literature is filled with successively lower power and lower area radios, some fabricated using CMOS technology, for high-rate communication at 60 GHz over distances of about 10 m. [37]

Given the zig-zag dipole result mentioned earlier, it seems clear that a 60 GHz antenna could fit within 1 mm^3. Given the rate at which 60 GHz transceivers are shrinking, due to the intense interest in 7 GHz of unlicensed spectrum, it seems likely that in the coming years we will see a complete 60 GHz radio, including antenna, that can fit inside 1 mm^3. This kind of radio would be useful for systems with very sparse paint nodes, separated by many meters.

4.4 CPU and Memory

4.4.1 CPU

The V1.0 lab-scale paintable computer described in this report uses the Atmel ATFR40162 processor, which has an ARM7TDMI core. This is a 32-bit RISC integer machine.

On a 130 nm process, the ARM7TDMI fits inside 0.26 mm^2 of silicon area, and consumes **60 µW/MHz**, with an instruction rate of 0.9 MIPS/MHz, and runs at up to 133 MHz. [38] This core is fully static, so it can be run at lower speeds with a proportional decrease in power.

4.4.2 Static RAM

A static RAM cell has six transistors, consumes **0.57 μm^2** of silicon area on a 65 nm process [39], and 2.0 μm^2 of silicon area on a 130 nm process. [40]

The 256K x 8 SRAM array in the V1.0 paint particle requires about 2 x 10^6 cells, for a total area of 1.1 mm^2 on a 65 nm process and 4 mm^2 on a 130 nm process.

For the construction of a full static machine with a reasonable amount of RAM, SRAM area is much more important than processor area. For a 1 mm^2 computer fabricated on a 65 nm process, a 32-bit processor will easily fit, but only 64K x 8 of SRAM will fit.

If a larger amount of RAM is needed, DRAM or 3D integration could be potentially be used.

4.4.3 FLASH

FLASH memory (or mask ROM) will be required for operating system storage by paint particles, and may also be useful for long-term zero-power storage of information, for example, in the "black paint" holistic data storage application.

FLASH memory requires much less area per bit than SRAM; **0.16 μm^2** per cell on a 130 nm process. [41] A 256K x 8 FLASH memory array requires 0.32 mm^2. While the industry is confident that FLASH memory will eventually scale down to 45 nm processes and beyond, [42] at the time of writing, this is appears to be the smallest published FLASH cell.

4.5 Light Emitting Diodes

4.5.1 Power Requirements

In this section, we determine how much electrical power is needed, per particle, to get a display with a particular brightness in Nits. Typical display brightness values are 200 Nits for indoor-readable LCD monitors and 1500 Nits for a sunlight-readable displays.

To get information on the efficiency of light-emitting diodes, we obtained datasheets from Lumileds Corp. on high-efficiency red, green, and blue LED's available in bare die form. This data is reprinted below.

Table 4. Electro-Optical Efficiency Information from LED Data Sheets.

Color	Process	Sample Device	Data
Red 626 nm	TSAlInGaP	HWFR-B517 (Lumileds)	0.42 lm @ 40mA, 2.3 V
Green 520 nm	InGaN	HWFR-P5G2 (Lumileds)	4.4 lm @ 50 mA, 4.0 V 2.2 lm @ 20 mA, 3.2 V
Blue 475 nm	InGaN	HWFR-P5B2 (Lumileds)	1.6 lm @ 50 mA, 4.0 V 0.8 lm @ 20 mA, 3.4 V

According to the Lumileds InGaN data sheet, bare die LED's emit light over 2.6 steradians of solid angle. From this information, we calculated the luminous intensity of each LED in candela. Also, by multiplying the stated voltage times the stated current, we calculated the electrical power consumed at the given conditions.

Table 5. Calculated Electro-Optical Efficiency Information

Color	Luminous Flux at Test Condition	Luminous Intensity at Test Condition	Electrical Power at Test Condition	Efficiency at Test Condition
Red 150 lm / W	0.42 lm	160 mcd	92 mW	3.0%
Green 500 lm/W	2.2 lm	846 mcd	64 mW	6.9%
Blue 100 lm/W	0.8 lm	308 mcd	68 mW	11.8%

Table 6. Electro-Optical Efficiency of RGB LED's

Color	Electrical Power per Luminous Intensity
Red	588 mW / candela
Green	76 mW / candela
Blue	222 mW / candela
Total (RGB White)	886 mW / candela

Based on this information, we developed the following formula to give the power requirements for an LED display made with red, green, and blue LED's:

$$P = \left(0.886 \quad \frac{W}{candela}\right) \cdot \left(A \quad m^2\right) \cdot \left(B \quad \frac{candela}{m^2}\right) \qquad \textit{Equation 38}$$

where B is the brightness of the display in Nits (candela/m^2), and A is the area of the display in m^2.

From Table 5, we can see that the efficiency of LED's is between 3% and 12%, so most of the power predicted by this formula is dissipated as heat, and only a small fraction leaves the particle as photon flux. This means that heat dissipation can place a limit on the minimum particle size for a display of a given brightness and particle spacing.

Table 7. LED Power Requirements

Display	Power Required
LED monitor 30 cm square box 200 nits brightness	17 W
Sunlight Readable Full-Color Paintable Display (per node) 0.25 mm^2 display area per particle 1500 nits brightness	**325 μW**
Indoor Readable Full-Color Paintable Display (per node) 0.25 mm^2 display area per particle 200 nits brightness	**43 μW**

4.6 Integration of Dissimilar Process Technologies

The example application (display) that is the main focus of this report requires the integration of different process technologies: CMOS, for logic and communications, and one or more III-V processes, for visible LED's of various colors.

There are several excellent candidate solutions for batch, wafer-scale combination of these processes. They are Wafer-to-wafer bonding [43], III-V fabrication on top of CMOS through SiGe virtual substrates, [44] and epitaxial lift-off. [45]

4.7 Economics

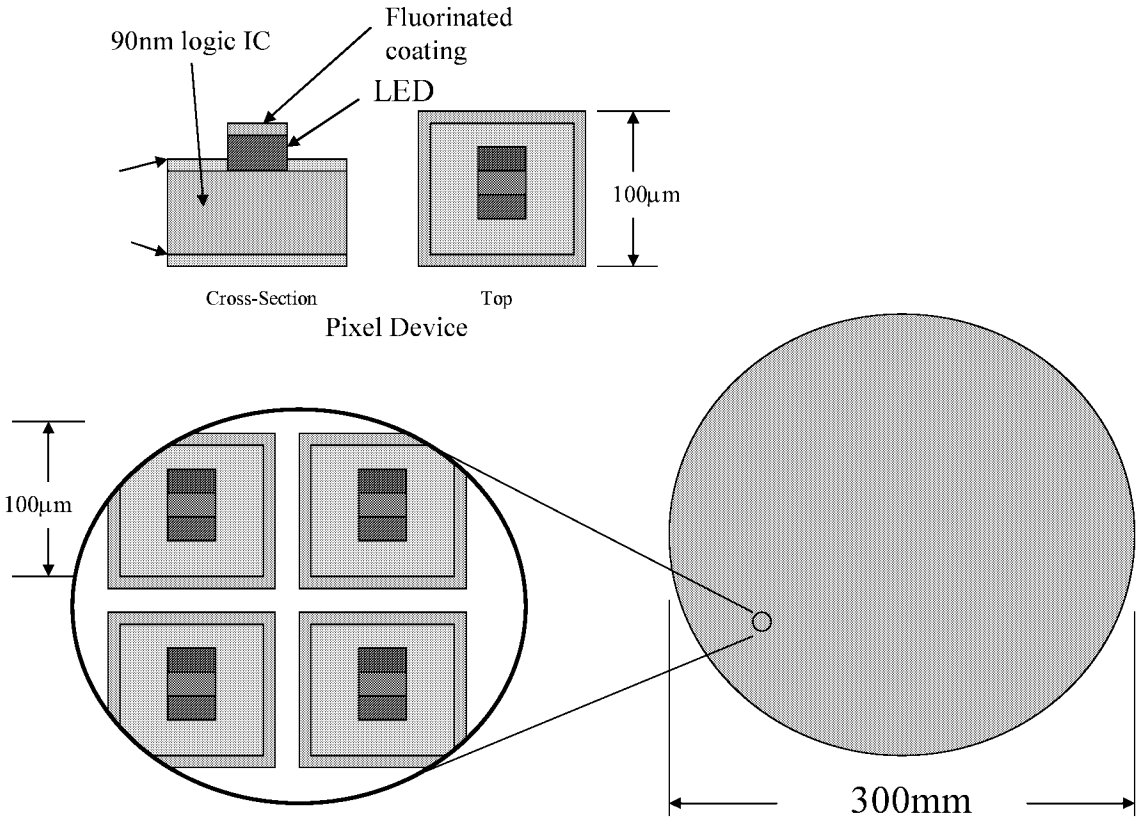

Figure 22. **Batch Fabrication of Display Particles**

The cost of silicon wafer fabrication, for the leading edge logic technology, has remained approximately constant at $\$16/in^2$ for the past 20 years. [46] This is approximately 2.5 cents per mm^2. The yield of a semiconductor process goes up with decreasing device size; 1 mm devices can be fabricated with near-unity yield on 300 mm wafers. [8]

The cost of electronic devices sold on the market comes from the cost of packaging, test, assembly, engineering, overhead, and profit; very little of it is due to the wafer fabrication process itself.

Ordinarily, test is a very important step; but the paintable computing architecture can tolerate some defective nodes. [8]

Here is a process flow for the manufacturing of paintable display particles:

1. Fabricate a silicon wafer with deep-submicron CMOS digital logic.

2. Grow SiGe virtual substrates and fabricate III-V LED's on top of the same wafer.

3. Add any other required materials; magnetic cores for power conversion, for example.

4. Coat the wafer with a protective layer to protect the devices from mechanical damage.

5. Dice (singulate) the wafer by deep reactive ion etching.

6. Mix the resultant millions of dice with the paint binder.

7. Paint onto a surface.

The cost of these process steps is given by the cost of the equipment and labor, multiplied by the process time, plus the cost of raw materials.

Fabrication of CMOS wafers is a complicated process involving many steps, and highly skilled labor, and highly expensive equipment. Thus, it is reasonable to say that the cost of steps 2-7 should not be out of line with the cost of step 1.

For particles without LED's, we estimate the fabrication cost at 1.5 the cost of deep-submicron CMOS fabrication, **3.75 cents / mm^2**. For particles with LED's, we estimate the fabrication cost at 2 times the cost of deep-submicron CMOS fabrication, **5 cents / mm^2**.

5 V3.0 PAINTABLE DISPLAY SYSTEM DESIGN

In this section, we present the top-level design of a paintable display.

This display has the same resolution and area as a 640x480, 17" diagonal, 4:3 aspect ratio computer monitor. But because it is a paintable display, it could also be painted in the form of a 1" x 150" strip, or painted onto the surface of a sphere.

Because it turns out to be too expensive to put a general-purpose processor behind every pixel, we propose to use two kinds of particles; 1 mm particles including a processor and memory, called *rendering particles,* and smaller 110 μm *display particles,* containing light-emitting diodes, a communications receiver, and minimal logic.

5.1 Display Particles

This display has 307,200 pixels, each responsible for 0.25 mm^2 of display area. From §4.5.1, 43 μW LED power are required for an indoor-readable display, and 325 μW are required for a sunlight readable display.

Each display particle needs to receive 24 bits of color information from a rendering particle at a rate of 60 Hz. Allowing for 3 times this number of bits, to account for warm-up time, identification bits, and clock skew, display particles need to receive data at 4.3 kb/s.

Because the paint binder needs to contain specialty power-supply materials, it is unlikely to be transparent. Therefore, we select inductive communication over optical communication.

A detailed design example for near-field inductive communication is given in §4.3.2.2, between two particles with 1 mm^2 coils. Here, we consider communication between a 1 mm^2 rendering particle and a display particle 10 times smaller in linear dimension. Referring to Equation 29, to keep the same mutual inductance, we need to reduce the communications radius by a factor of 4.7, from 10 mm to 2 mm, to be able to use the results of §4.3.2.2.

The display particles can receive data with an efficiency of better than 37 pJ/bit.
At the data rate above, this means 0.16 μW per display particle is required for communications.

The digital circuitry in a display particle is shown in Figure 23. We estimate 4 μW for the oscillator and 1 μW for the remainder of the digital logic, which runs at the low speed of 4.3 kHz.

Adding up the power for display, communication, and logic, we get a power budget of **50 μW** for indoor-readable display particles and **336 μW** for sunlight-readable display

particles. The heat dissipation limit at 100 μm particle size is about 400 μW; so indoor-readable particles will run cold, and the sunlight-readable particles will run hot.

The 307,200 display particles required consume 3,072 mm^2 of total silicon area. Using the cost estimate for particles with III-V integration given in §4.7, we can estimate the cost for the display particles at **$154**.

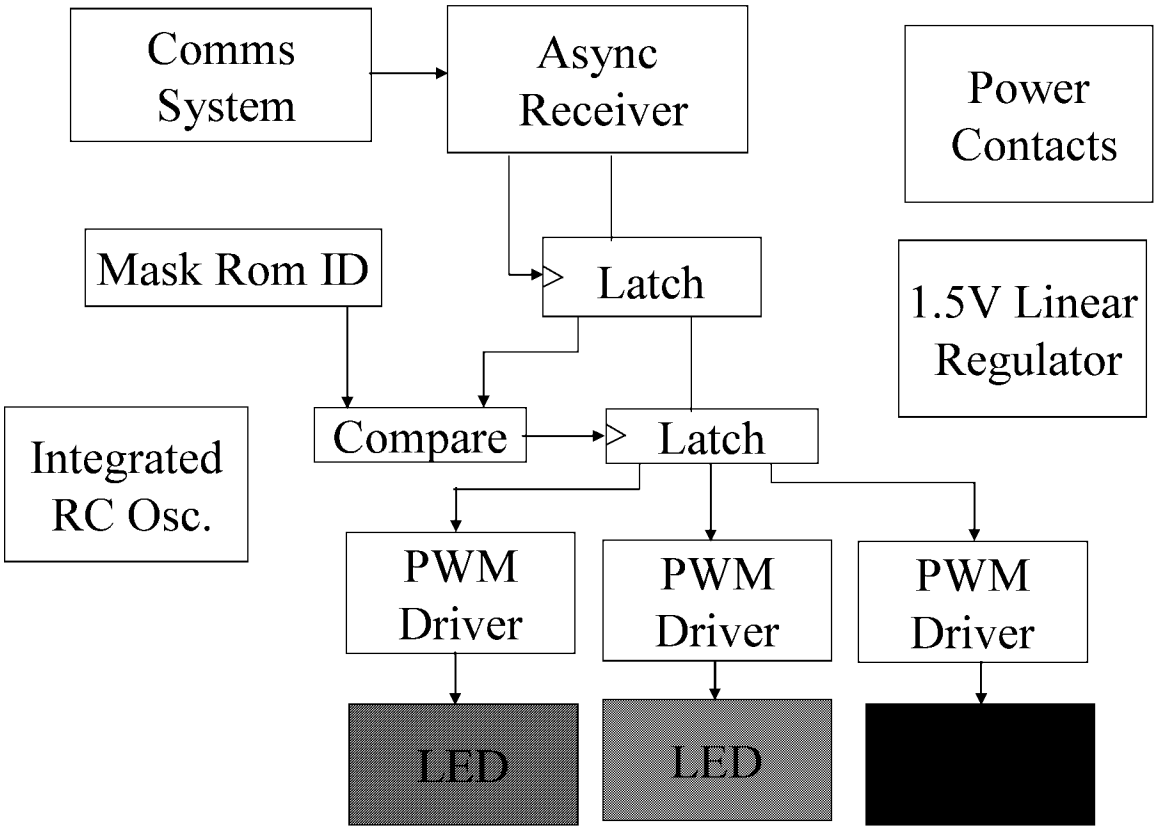

Figure 23. Display Particle Block Diagram

In the proposed paintable display, there are two kinds of particles: 1 mm rendering particles, which have a processor and memory, and 10-μm display particles, each of which has a tricolor LED, an inductive communications system, and simple digital decoding circuitry.

5.2 Rendering Particles

We specify 5,120 rendering particles per display, so that there will be one rendering particle every 3 mm of linear dimension. (With a very uniform coating process, fewer may be required; with a very haphazard coating process, more would be required.) At the density stated, there is one rendering particle for every 60 display particles.

Rendering particles will need to collaboratively decode MPEG streams and postscript files. The main computational task in MPEG decoding is computing 8x8 inverse discrete cosine transforms, to render 64 pixels of the image. (Each rendering particle is

responsible for approximately that number of pixels.) Performing 8x8 IDCT's at 60 Hz takes 35,000 instructions per second. Data transfer and operating system overhead are another computational task; since we cannot accurately estimate this overhead, we will be conservative and guess that processor will need to run at 1 MHz and have 16 KB RAM. Then, rendering particles will draw **60 µW** each.

The 5,120 rendering particles consume 5,120 mm^3 of silicon area. Together, they have 90 MB of static RAM and can execute 10^9– 10^{11} operations per second.

Using the cost estimate for paint particles without III-V integration, given in §4.7, we can estimate the manufacturing cost of these particles at **$192**.

5.3 Power

Inside each mm^2 of display area, there are four display particles and 1/9 of a rendering particle. The system power consumption per mm^2 is

50 µW x 4 mm^{-2} + 60 µW x 0.11 mm^{-2} = **207 µW / mm^2** (Indoor Readable)
336 µW x 4 mm^{-2} + 60 µW x 0.11 mm^{-2} = **1.4 mW / mm^2** (Sunlight Readable)

5.4 Power: Random Environment

Using the inside-out zinc-air battery concept detailed in §4.2.2.1, and assuming a 1 mm thick paint layer, 6.0 J/mm^2 of energy is available from battery reactants stored in the binder. A painted primary battery would allow **eight hours** of indoor operation and **71 minutes** of outdoor operation, for a display of any size.

5.5 Power: Controlled Environment

To power the display from the commercial power grid of or from a vehicle electrical system, we can use reactive power transfer, described in §4.2.3.2.

Reactive power transfer can deliver the required power at 8 MHz with $V_{DD(max)}$ = 3.3 V, or at 1 MHz with $V_{DD(max)}$ = 10 V. A process capable of a least 5 V will be required to fabricate the display in any event, since III-V LED's require that much turn-on voltage.

The oscillating electric field required for reactive power transfer must be produced directly under the display. This could be done by patterning a single-layer flex circuit with interdigitated electrodes at 100 µm pitch, a pitch which is readily achievable through low-cost patterning methods.

CONCLUSIONS AND RECOMMENDATIONS

Paintable displays appear to be feasible. Battery-powered, field-paintable displays would have a battery life in the 8-hour range. Factory-coated displays could use external power. Both variants could be made at a cost competitive with medium-size LCD's on an area basis, but could be scaled to any size. In addition, they could conform to unusually shaped 3-D surfaces, and could flex. Applications other than displays, such as "black paint" holistic data storage, and very dense sensor networks, also appear to be feasible. The high computing throughput of such a system, which constructed with the paint architecture, would also allow the integration of simulation, modeling, inference, and/or rendering capabilities *in situ.*

We explored the technical feasibility of building paintable systems. We constructed and programmed a lab-scale prototype system. Using self-assembling code, we programmed the system to act as a display. Working together, the particles render and display a postscript-format image file. In addition, we constructed a functional prototype demonstrating power distribution to and operation of randomly oriented millimeter-scale semiconductor devices.

We performed a series of basic engineering calculations to determine the feasibility of paintable systems with 1 mm^3 paint particles. Particles can dissipate 10 mW heat, generate 6 J of electricity from internal zinc-air batteries or 1.5 J from internal combustible fuel. Photovoltaic cells provide 300 μW outdoors and 3.0 μW indoors. Paintable systems can store battery reactants in the paint binder; 6 J / mm^3 of binder can be stored, and diffusion is fast enough to transport the reactants to the particles. Reactive power transfer is an efficient method to transfer power to sparse, randomly placed particles. The available power from reactive transfer is proportional to V_{DD}^2: 100μW at 3.3V and 12 mW at 35V. Inter-particle communications is possible via optical, near-field, and far-field electromagnetic systems. Optical systems allow communication with very low area (sub-mm) particles, and 24 pJ/bit. Near-field electromagnetic gives precisely controlled neighborhoods, localization capability, and 37 pJ/bit. Far-field radio communication between widely spaced particles may be possible at 60 GHz; antennas that fit inside 1 mm^3 exist; complete transceivers do not. A 32-bit CPU uses less than 0.26 mm^2 die area, 256K x 8 SRAM uses 1.1 mm^2, and 256K x 8 FLASH uses 0.32 mm^2. III-V LED's may be fabricated on Si wafers using SiGe virtual substrates.

We selected technologies for a 17" diagonal, 640 x 480, paintable color display application. We propose to use a mixture of two kinds of particles: 1.0 mm rendering particles, with a microprocessor and memory, and 110 μm display particles, with simpler circuitry. Display particles use 50 μW for indoor-readable brightness and 336 μW for outdoor-readable brightness. Storage of zinc-air battery reactants in the paint binder allows 8 hour battery life for indoor use and 1 hour battery life for outdoor use. Reactive power distribution allows continuous operation from external power. **The 300,000 paint particles required for this display could be manufactured for about $350.**

If a program were to be initiated in this area, there would be many specific technical risks. The most significant of these risks are enumerated below.

- Requirements of an actual design may differ from estimates. In this report, we estimated power, heat, and communications requirements using simple formulas and approximations. In doing so, we included some safety factors and "engineering margin." However, it is possible that once an actual design is sketched out, power or size requirements may be larger than what is calculated. This is a risk.

- Materials development may be difficult or time-consuming. In this report, we call out requirements for materials to enable power transfer to paint through the binder, or power storage in the binder. The requirements on these materials seem plausible, and do not contradict any well-known physical laws, but there may be detailed reasons why they cannot be developed or manufactured at a suitably low cost. This is a risk.

To enable this paintable display application, certain specific technical advances would need to be made. These are:

- Development of a Batch Packaging Process: A process to be applied to an entire wafer that would protect small, roughly cube-shaped semiconductor devices from damage after a subsequent singulation. A dielectric coating process may be appropriate.

- Development of a Batch Singulation Process: A process to singulate a wafer into millions of elements. Deep reactive ion etching, which is currently used to make macroscopic, lithographically defined holes in wafers, may be an appropriate process.

- Development of Suitable Binder Materials: For binder-stored power, a paint binder containing battery reactants and having the proper fluid and physical properties (e.g. density, viscosity, surface tension) required for painting would need to be developed. For reactive power transmission, a paint binder having the proper electrical properties (high dielectric constant) as well as physical properties for painting would need to be developed.

- Power System Development: Using the above-mentioned binder materials, design and fabrication of a system to transmit power efficiently to disconnected, randomly oriented millimeter scale devices. We began work, both theoretical and experimental, during this seedling phase; more work is needed.

- Design and fabrication of an actual-size paintable system.

Potential Future Work:

- Manufacturing systems using the paint architecture but with non-random geometries via wafer fabrication or roll-to-roll fabrication and roll-to-place batch assembly.

- Exploring a version of the self-assembling code programming model that uses a cellular-automata computing elements, rather than Von Neumann microprocessors. This might allow simpler processing elements, to further increase the granularity, density, and fungibility of the resulting systems.

- Developing actuated paint nodes that can move relative to one another and to their environment, for programmable matter applications

- Examining the feasibility of using a paint array as a programmable electromagnetic transducer, in which a system of paint particles operates together as a distributed communications system, amplifier, and antenna. If feasible, this could allow a system of paint particles to communicate with a communications satellite via RF.

- Building a prototype paintable system, such as a paintable display or paintable computer.

In summary, power and communications do present limits on what kinds of applications can be realized using the paintable computing architecture. Nonetheless, many interesting applications appear to be feasible, including paintable computers and displays with integrated rendering, at a large-volume manufacturing cost competitive with current practice.

This document reports research undertaken at the U.S. Army Research, Development and Engineering Command, Natick Soldier Center, Natick, MA, and has been assigned No. NATICK/TR-06/012 in a series of reports approved for publication.

6 REFERENCES

[1] Moore, G., *Cramming More Components onto Integrated Circuits*. Electronics, April 19, 1965.

[2] Moore, G., *No Exponential is Forever...but We Can Delay 'Forever.'* Presentation at the IEEE International Solid-State Circuits Conference, February 10, 2003

[3] Abelson, H., Allen, D., Coore, D., Hanson, C., Homsy, G., Knight, T., Nagpal, R., Rauch, E., Sussman, G., Weiss, R., *Amorphous Computing.* Communications of The ACM, 2000, Vol. 43, No. 5, pp. 74-82

[4] Butera, W. J. *Programming a paintable computer*. Massachusetts Institute of Technology Doctoral Thesis, 2002, UMI Order Number: AAI0804036.

[5] Lifton, J., Seetharam, D., Broxton, M., and Paradiso, J. A. Pushpin Computing System Overview: A Platform for Distributed, Embedded, Ubiquitous Sensor Networks. In *Proceedings of the First international Conference on Pervasive Computing* (August 26 - 28, 2002). F. Mattern and M. Naghshineh, Eds. Lecture Notes In Computer Science, vol. 2414. Springer-Verlag, London, 139-151.

[6] Warneke, B.A. Scott, M.D. Leibowitz, B.S. Lixia Zhou Bellew, C.L. Chediak, J.A. Kahn, J.M. Boser, B.E. Pister, K.S.J. An autonomous 16 mm/sup 3/ solar-powered node for distributed wireless sensor networks. Proceedings of IEEE Sensors, 2002, Volume 2, Page 1510-1515, ISBN 0-7803-7454-1

[7] Drzaic, P., Chiang, A., Stewart, R., Hermanns, A., Shi, Y., Jacobsen, J. *Plastic-film displays with NanoBlock IC drivers integrated by a fluidic self-assembly process.* Journal of the Society for Information Display, 2003, Volume 11, Issue 1, pp. 81-87

[8] Jacobson, J., Gershenfeld, N., Butera, B., *The V1.0 'Pushpin' Particle 2-D Self-Assembling Display Array.* Natick Technical Report No. *NATICK/TR-04/018, August 2004.*

[9] Mills, A.F. *Basic Heat and Mass Transfer*. 2nd Edition, Prentice Hall, 1998

[10] Tarascon, J.M., Armand, M. *Issues and challenges facing rechargable lithium batteries.* Nature, p.359, Volume 414, November 2001,

[11] Duracell. *Zinc-Air Technical Bulletin.* Boston: The Gillette Company, 2004.

[12] Bates, J.B., Gruzalski, G.R., Luck, C.F., *Rechargable solid state lithium microbatteries.* Proceedings of the 6th IEEE workshop on MEMS, Fort Lauderdale, FL, 1993, pp. 82-86

[13] Epstein, A.H, Senturia, S.D., Anathasuresh, G., Ayon, K. *Power MEMS and microengines*, Technical Digest of the International Conference on Solid-State Sensors and Actuators, 1997

[14] Madou, M. *Fundamentals of Microfabrication.* 2nd edition. CRC Press, 2002

[15] Blanchard, J., Henderson, D., Lal, A., *Nuclear Microbattery for MEMS Devices.* Project report submitted to U.S. Department of Energy

[16] MSDS, ^{63}Ni

[17] MSDS, ^{3}H

[18] Hall, E., *Radiobiology for the Radiologist.* 4th Edition, Lippincott Williams & Wilkins, 1993

[19] Shapiro, J., *Radiation Protection: a guide for scientists and physicians.* Cambridge: Harvard University Press, 1990

[20] Guo, H., Lal, A., *Nanopower Betavoltaic Microbatteries.* TRANSDUCERS: 12th International Conference of Solid-State Senors, Actuators, and Microsystems, Volume 1, p. 36-39, June 2003.

[21] Wang, S. Li, S., Du, Y., Xu, B., Li, L., Zhu, Y. Size Limit of Zinc Nanoparticles. The Preliminary Program for the Spring National Meeting on Engineered Particles or Engineered Nanoparticle Structures, Orlando F.L., 2006

[22] Paradiso, J. A. and Starner, T. 2005. *Energy Scavenging for Mobile and Wireless Electronics*. IEEE Pervasive Computing 4, 1 (Jan. 2005), 18-27.

[23] Virtuani, A.; Lotter, E.; Powalla, M.; Rau, U.; Werner, J.H. *Highly resistive Cu(In,Ga)Se2 absorbers for improved low-irradiance performance of thin-film solar cells.* Thin Solid Films, 2004, Vol. 451-452, pp. 160-165

[24] M. Green et. al, *Solar Cell Efficiency Tables (Version 18)*, Progress in Photovoltaics: Research and Applications, Volume 9, Wiley/Interscience, 2001

[25] Ogale, S.B. et. al, *High Temperature Ferromagnetism with a Giant Magnetic Moment in Transparent Co-doped $SnO_2\delta$.* Physical Review Letters, Vol. 91, No.7, August 2003

[26] Park, J., Allen, M. *Ultralow-Profile Micromachined Power Inductors With Highly Laminated Ni/Fe Cores: Applications to Low-Megahertz DC-DC Converters*. IEEE Transactions on Magnetics, Vol. 39, No. 5, September 2003

[27] Liang-Liang, X., Kumar, P.R., *A network information theory for wireless communications: scaling laws and optimal operation.* IEEE Transactions on Information Theory, Volume 50, Issue 5, pp. 748-767

[28] Lee, T.H, *CMOS RF: no longer an oxymoron.* Proceedings of the 19th Gallium Arsenide Integrated Circuit Symposium, pp. 244-247, 1997.

[29] Camilleri, N., Costa, J., Lovelace, D., Ngo, D., *Silicon MOSFETs, the microwave device technology for the 90s.* 1993 IEEE International Microwave Symposium Digest p. 545, 1993.

[30] Lee, T.H., The *Design of CMOS Radio-Frequency Integrated Circuits.* Cambridge: Cambridge University Press, 1998

[31] Leibowitz, B., Boser, B., Pister, K.S.J., *CMOS smart pixel for free-space optical communication.* Proceedings of SPIE – Volume 4306: Sensors and Camera Systems for Scientific, Industrial, and Digital Photography Applications II, May 2001, pp. 308-318

[32] Sunderarajan, M., Hershenson, M., Boyd, S., Lee, T.H., *Simple Accurate Expressions for Planar Spiral Inductances.* IEEE Journal of Solid-State Circuits, Vol. 34, No. 10, October 1999

[33] Lee, Y., *Antenna Circuit Design for RFID Applications.* Microchip Technology Corp, Chandler, AZ, AN710, 2003

[34] Balanis, C.A., *Antenna Theory: Analysis and Design.* 2nd Edition, John Wiley and Sons, 1996

[35] Lin, J., Sugavanam, A., Guo, X., Li, R., Brewer, J., Kenneth, K.O., *Integrated Antennas on Silicon Substrates for Communication Over Free Space.* IEEE Electron Device Letters, Vol. 25, No. 4, April 2004

[36] Porret, A.S., Melly, T., Python, D., Enz, C.C., Vittoz, E.A. *An ultralow-power UHF transceiver integrated in a standard digital CMOS process: architecture and receiver.* IEEE Journal of Solid State Circuits, Volume 36, Issue 3, pp. 452-466, March 2001

[37] Doran, C., Emami, S., Sobel, D., Niknejad, A., Broderson, R., *Design Considerations for 60 GHz CMOS Radios.* IEEE Communications Magazine, pp. 132-140, December 2004.

[38] Advanced RISC Machines Corp., *ARM7TDMI: ARM 32-bit RISC core with 16-bit system costs.* http://www.arm.com/products/CPUs/ARM7TDMI.html

[39] M. Bohr, et. al, *A 65 nm Logic Technology Featuring 35 nm Gate Lengths, Enhanced Channel Strain, 8 Cu Interconnect Layers, Low-k ILD and 0.57 μm^2 SRAM Cell.* Proceedings of the IEEE International Electron Devices Meeting, San Francisco, CA, 2004

[40] M. Bohr, et. al, *A 130nm Generation Logic Technology Featuring 70 nm Transistors, Dual Vt Transistors, and 6 layers of Cu Interconnects.* Proceedings of the IEEE International Electron Devices Meeting, San Francisco, CA, 2000

[41] Keeney, S., *A 130nm Generation High Density Etox™ Flash Memory Technology.* Technical Digest of the IEEE International Electron Devices Meeting, Washington D.C., pp. 2.5.1-2.5.4, Washington, DC, 2001

[42] Atwood. G., *Future Directions and Challenges for ETox Flash Memory Scaling.* IEEE Transactions of Device and Materials Reliability, Vol. 4, No.3, September 2004

[43] Gutmann, R.J., Lu, J.Q., Kwon, Y., McDonald, J.F., Cale, T.S. *Three-dimensional (3D) ICs: a technology platform for integratedsystems and opportunities for new polymeric adhesives.* First International IEEE Conference on Polymers and Adhesives in Microelectronics and Photonics, pp. 173-180, Potsdam, Germany, 2001

[44] Yang et. al, *Monolithic integration of III-V optical interconnects on Si using SiGe virtual substrates.* Journal of materials science: Materials in electronics, Vol. 13, Issue 7, Page 377, 2002

[45] Pollentier, I. et. al, *Epitaxial lift-off GaAs LEDs to Si for fabrication of opto-electronic integrated circuits*, IEEE Electronics Letters, Vol 26, No. 3, pp. 193-4, February 1990

[46] Personal Communication with David Tennenhouse, who is the Director of Research for Intel Corporation, September 2004.